

1. WORD COUNT

```
from pyspark.sql import SparkSession

# Create a SparkSession
spark = SparkSession.builder.appName("WordCount").getOrCreate()

file_path = "/content/FuelConsumption.csv"
text_file = spark.read.text(file_path).rdd.map(lambda r: r[0])

# Split the lines into words and count them
word_counts = text_file.flatMap(lambda line: line.split(" ")).count()

# Print the word count
print("Number of words in the file:", word_counts)

# Stop the SparkSession
spark.stop()
```

Number of words in the file: 3350

2 . VIDEO STREAM

```
pip install opencv-python

Requirement already satisfied: opencv-python in /usr/local/lib/python3.10/dist-packages (4.10.0.84)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.10/dist-packages (from opencv-python) (1.26.4)

import cv2
from google.colab.patches import cv2_imshow # Import cv2_imshow for Colab compatibility

# Replace with the path to your video file
video_file_path = "/content/demo_vid_1 (1).avi"

# Initialize video capture with the video file
video_capture = cv2.VideoCapture(video_file_path)

# Check if the video stream is opened successfully
if not video_capture.isOpened():
    print("Error opening video stream or file")
    exit()

while True:
    # Read a frame from the video stream
    ret, frame = video_capture.read()

    # If frame is read correctly, ret is True
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break

    # Perform analysis on the frame here
    # Example: Convert to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Display the resulting frame using cv2_imshow for Colab
    cv2_imshow(gray) # Replaced cv2.imshow with cv2_imshow

    # Exit if 'q' is pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

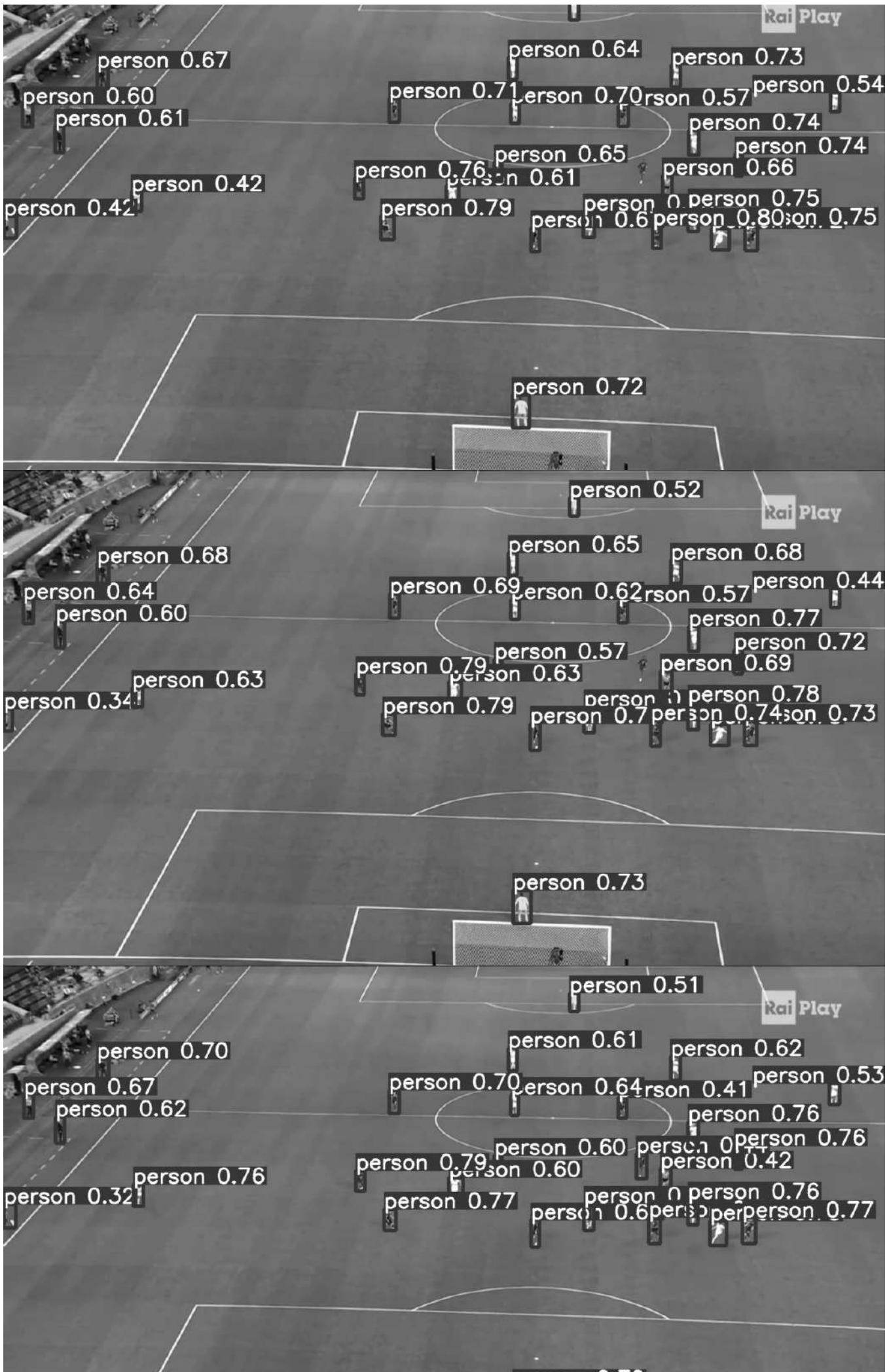
    # When everything done, release the capture
video_capture.release()
cv2.destroyAllWindows()
```

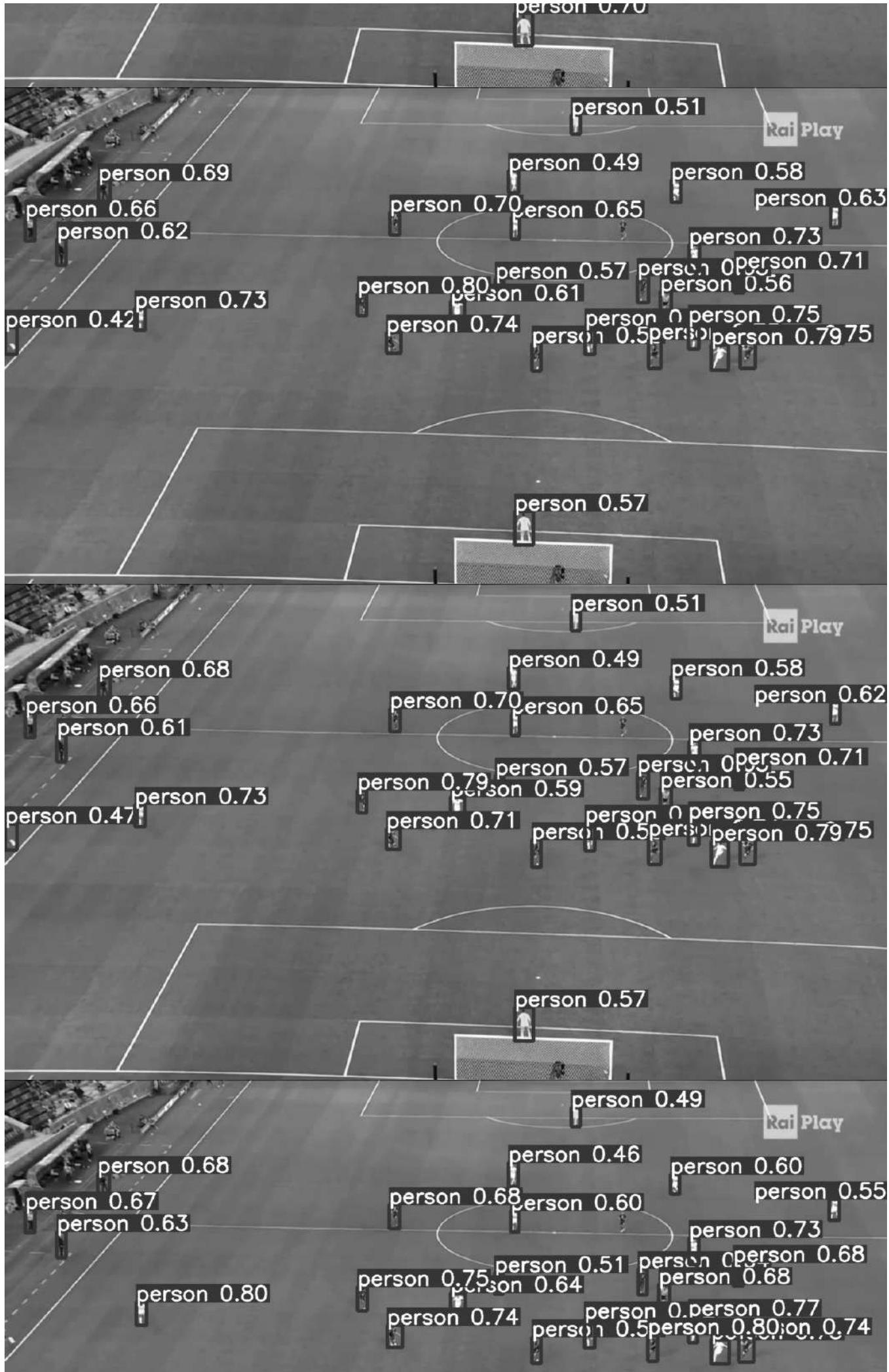


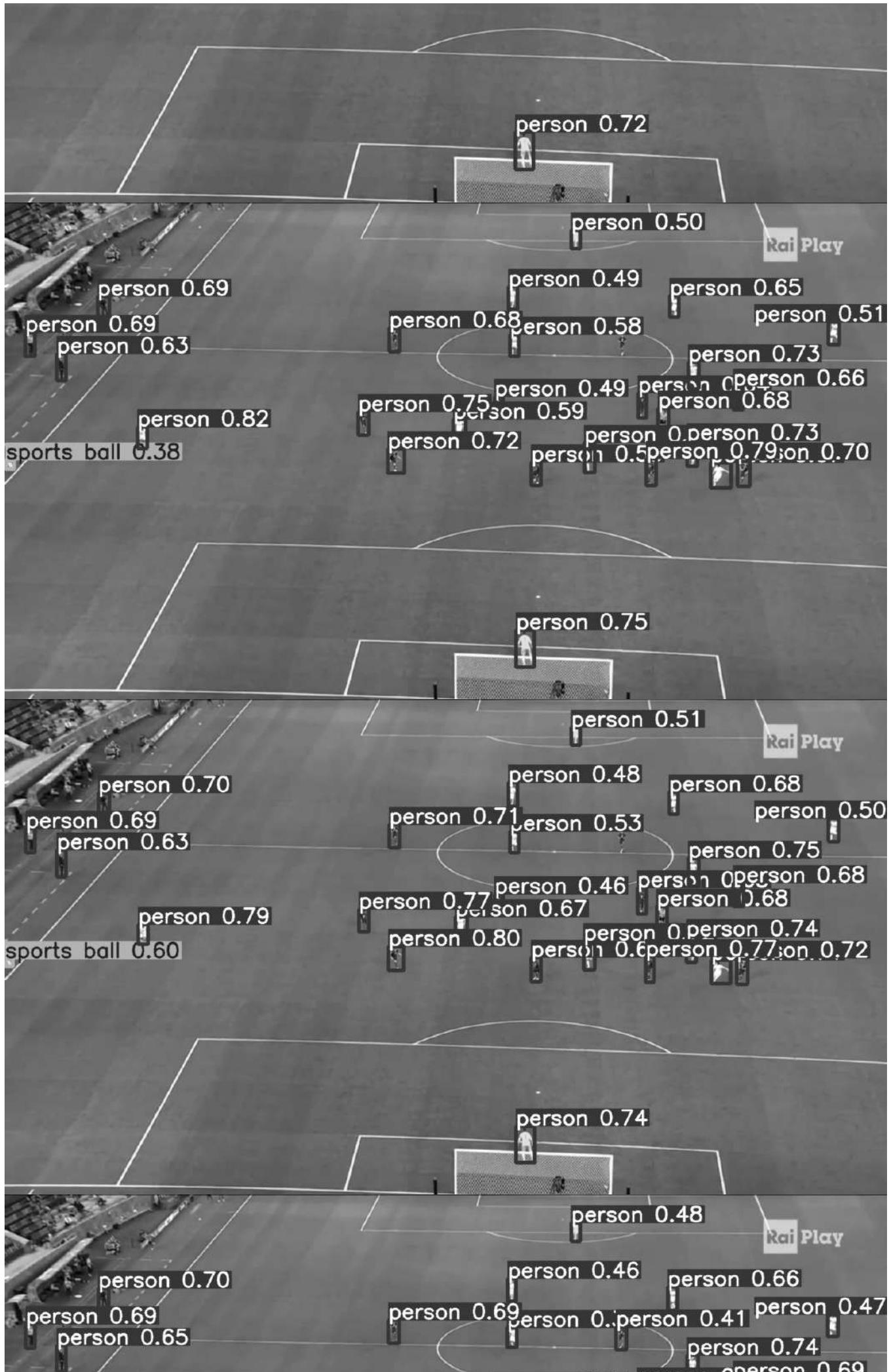


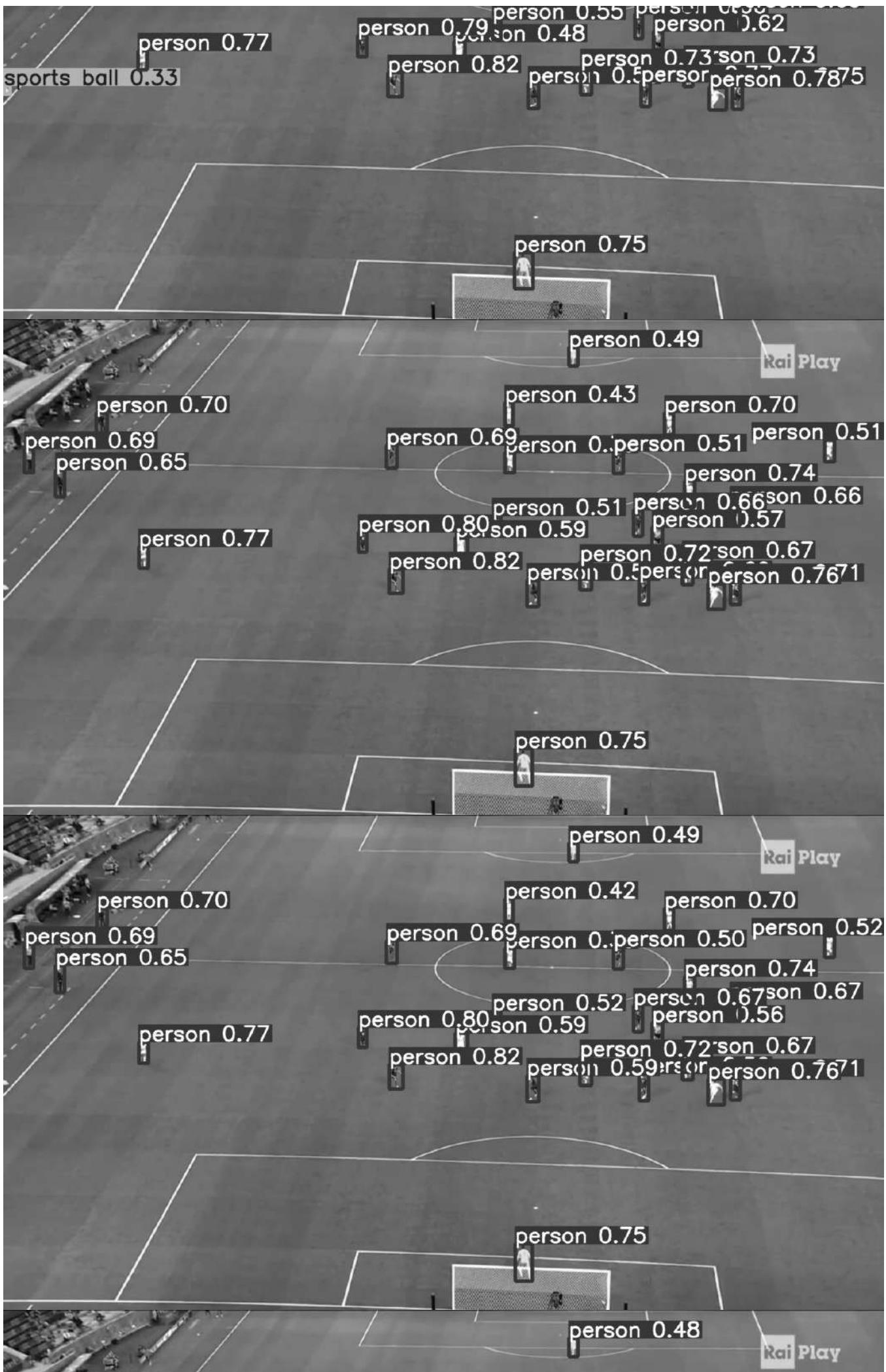




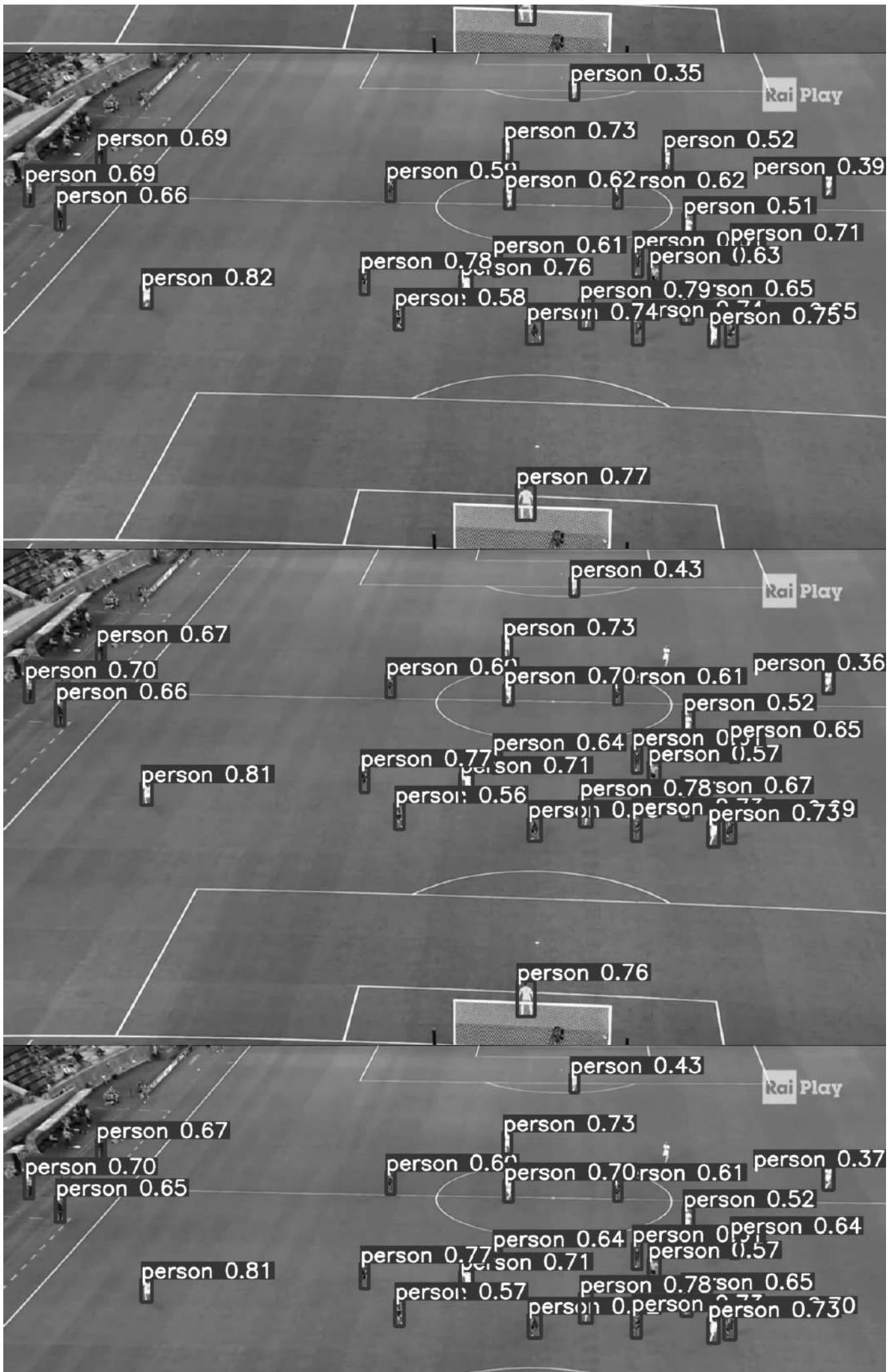


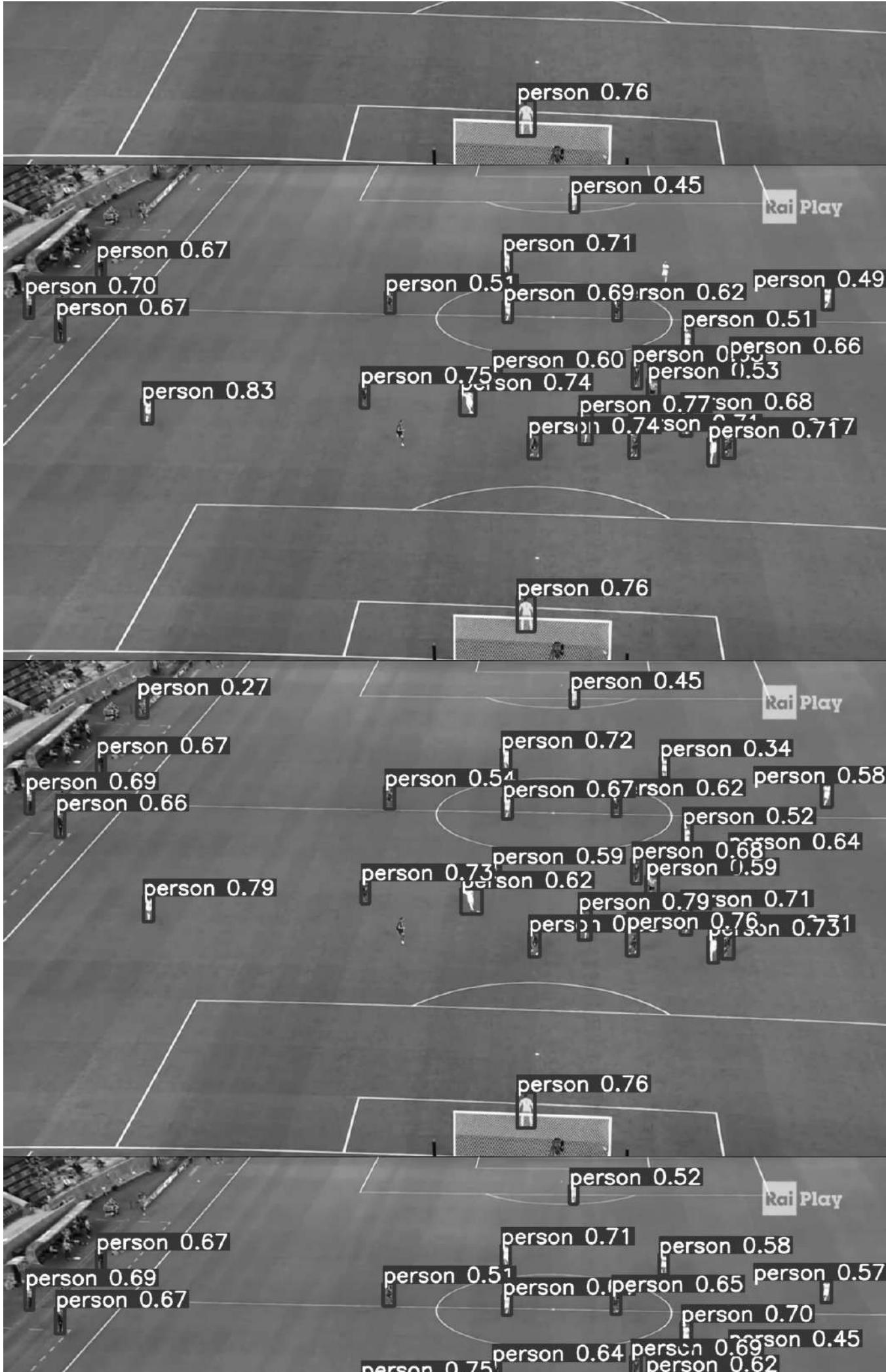


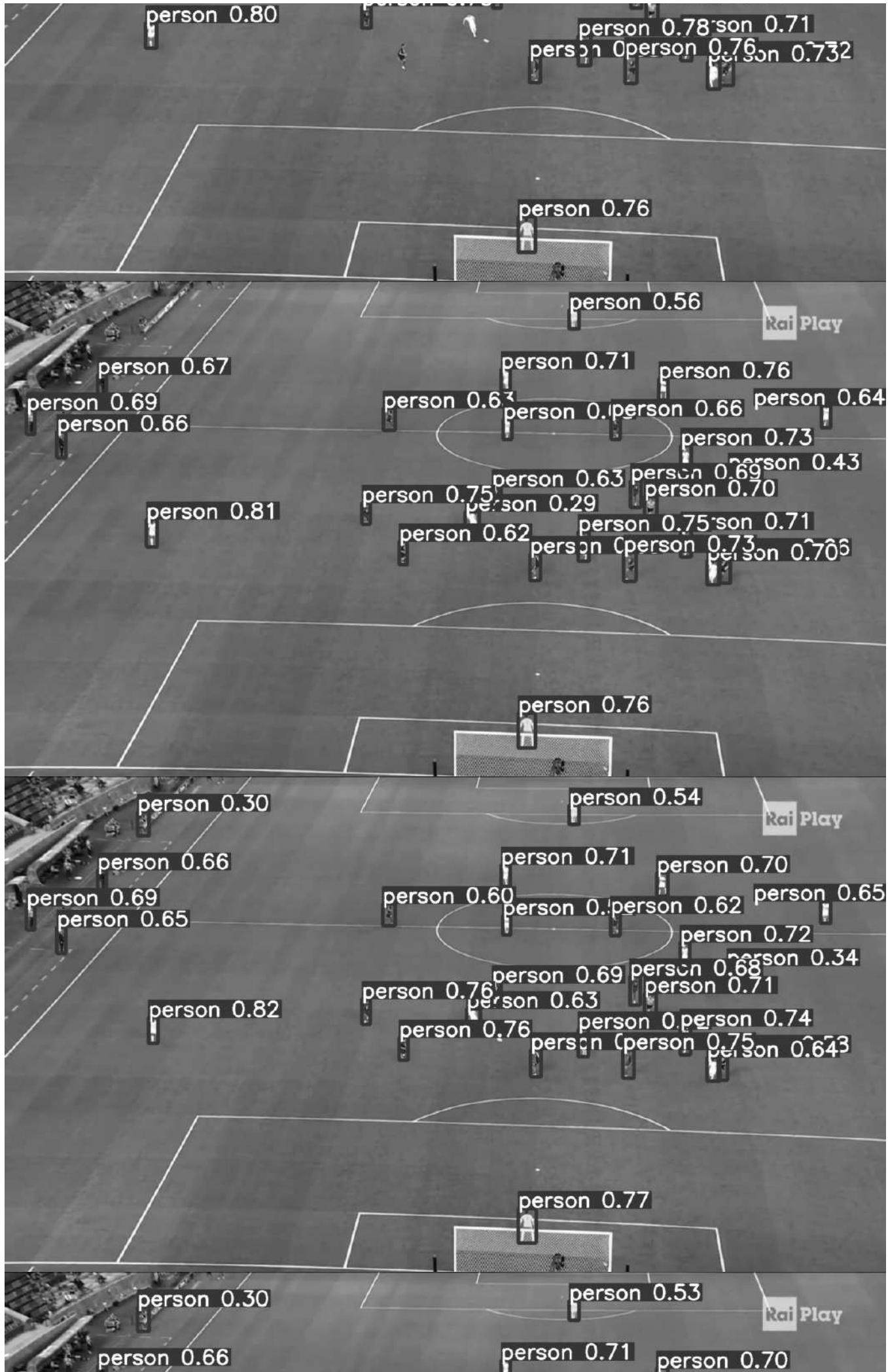


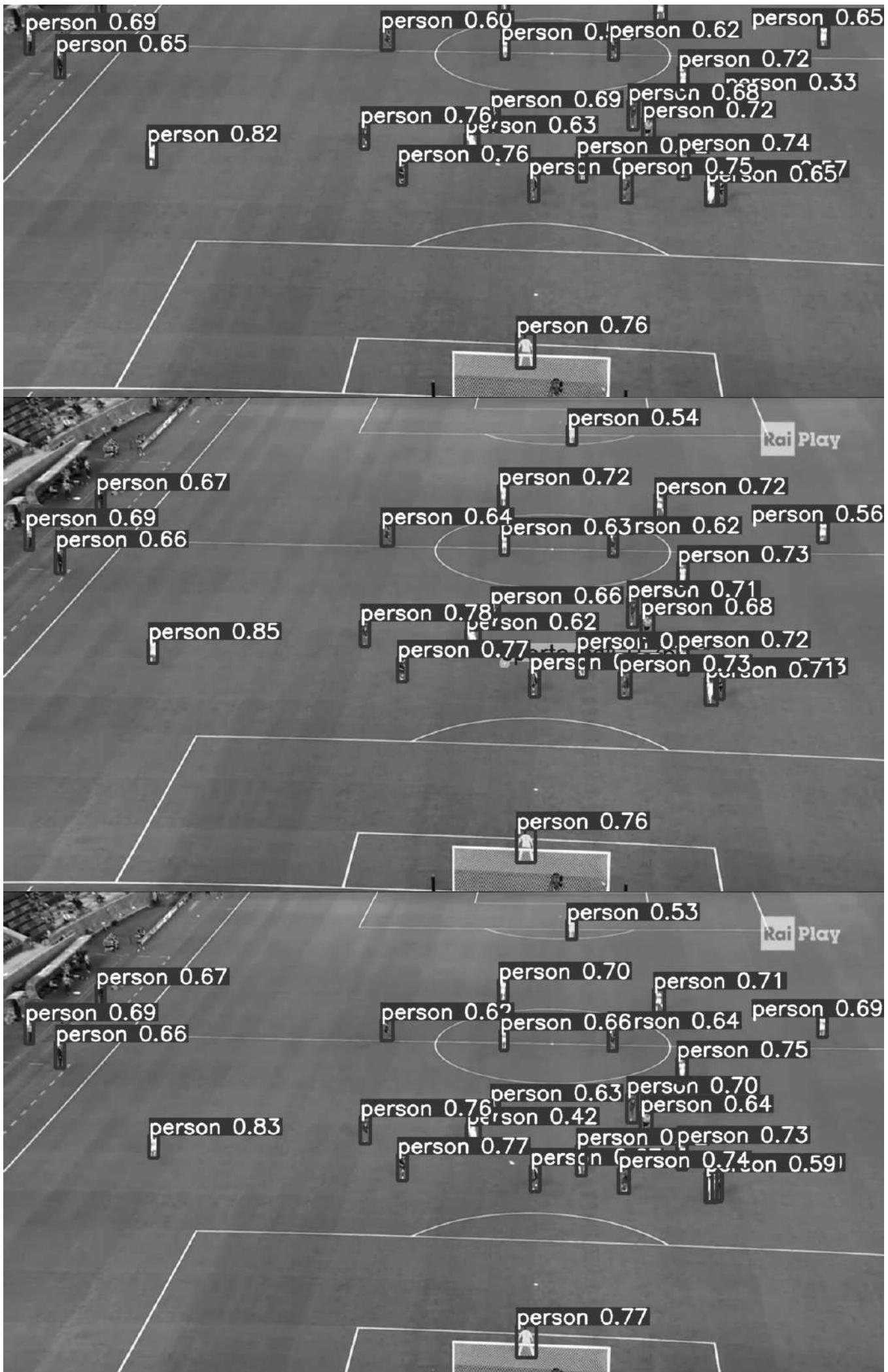


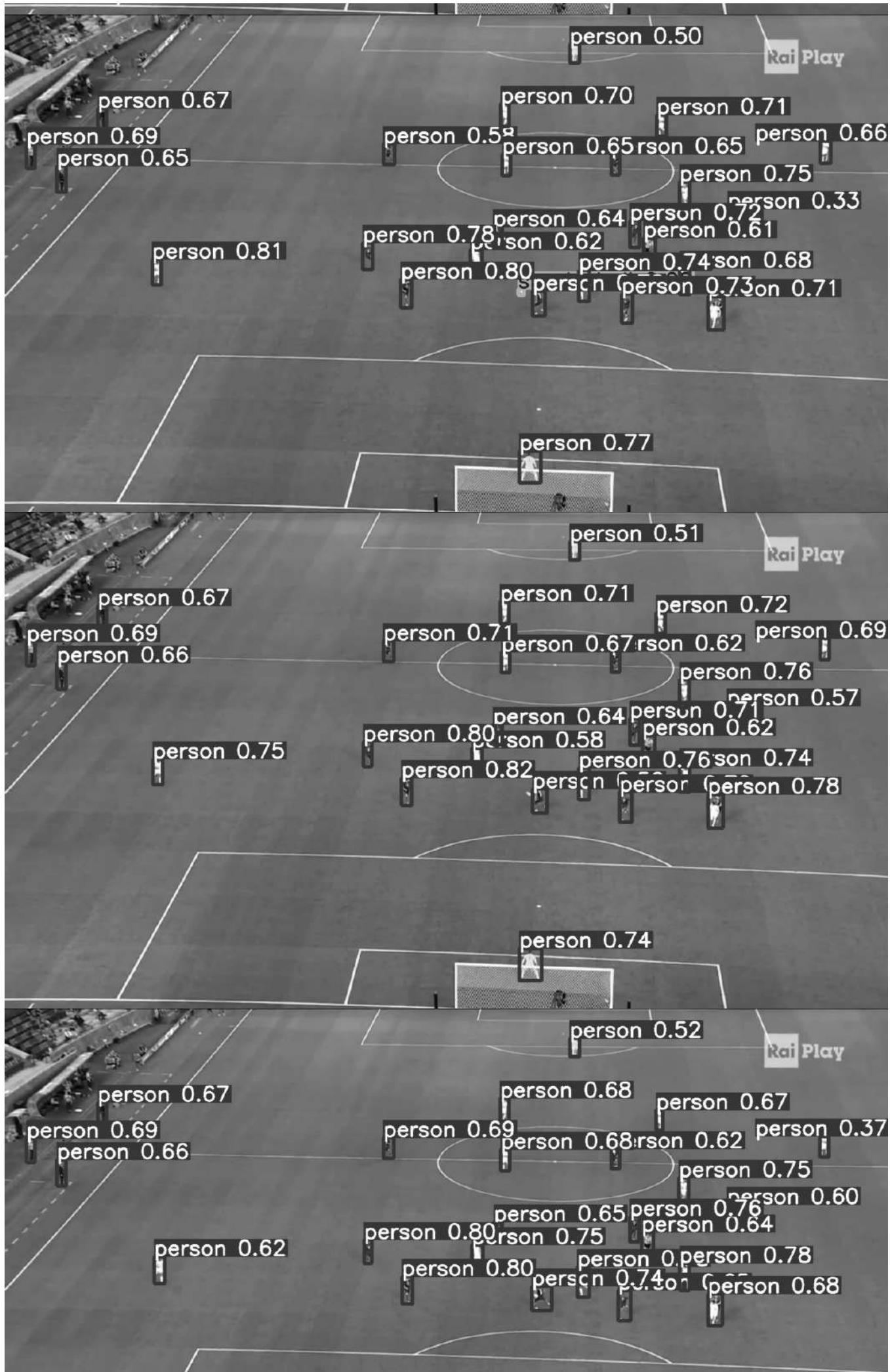


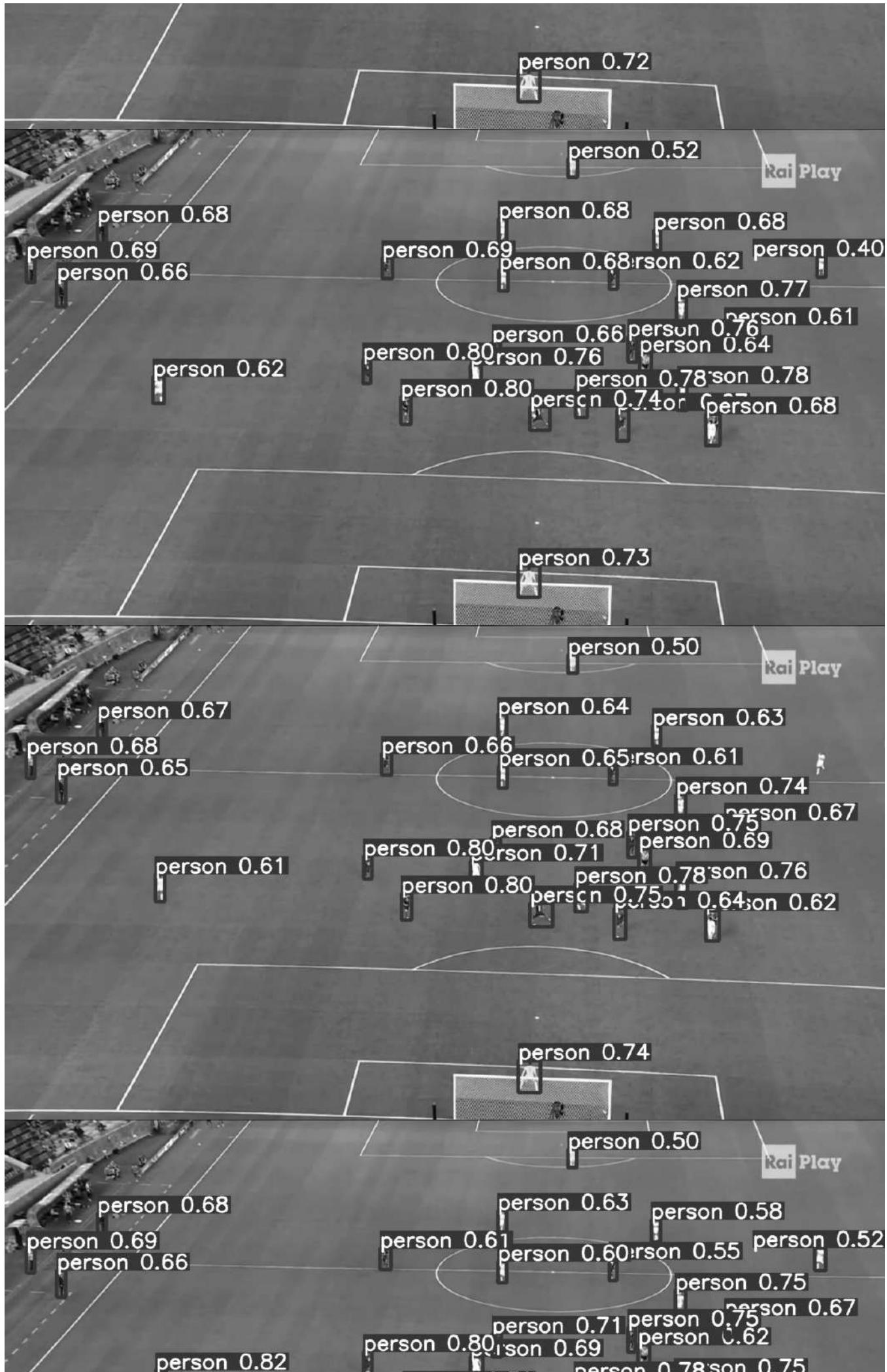


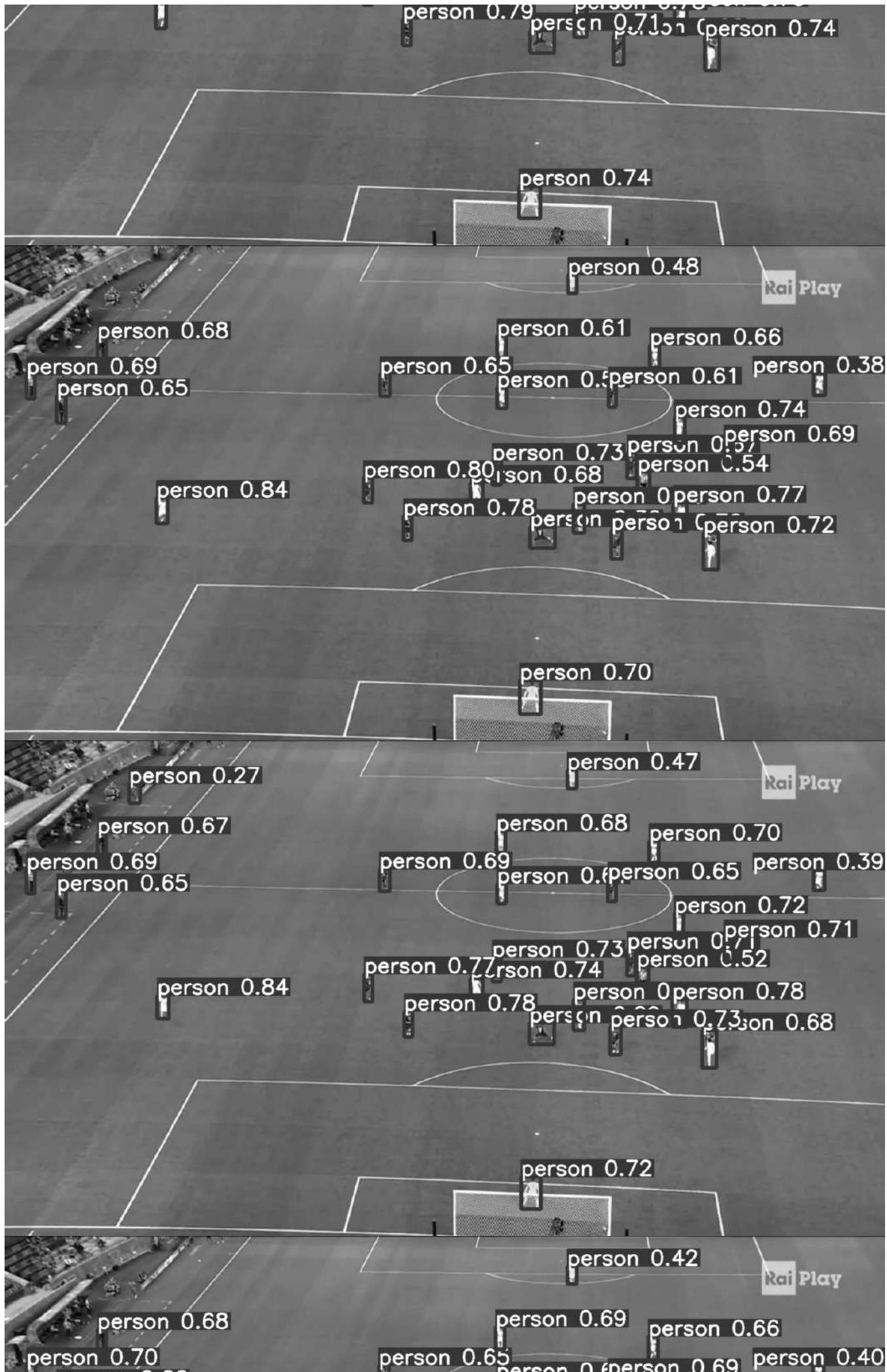


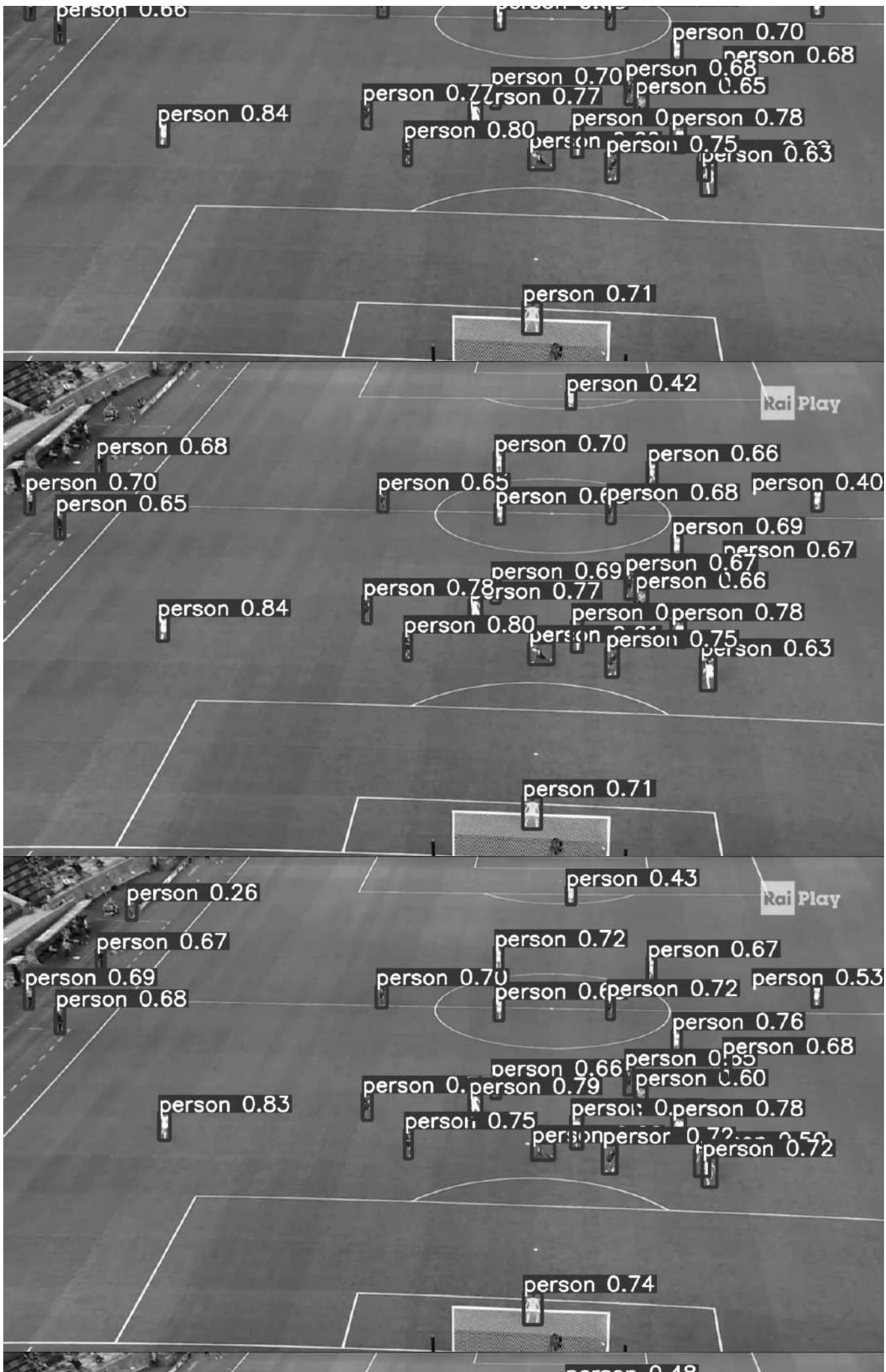


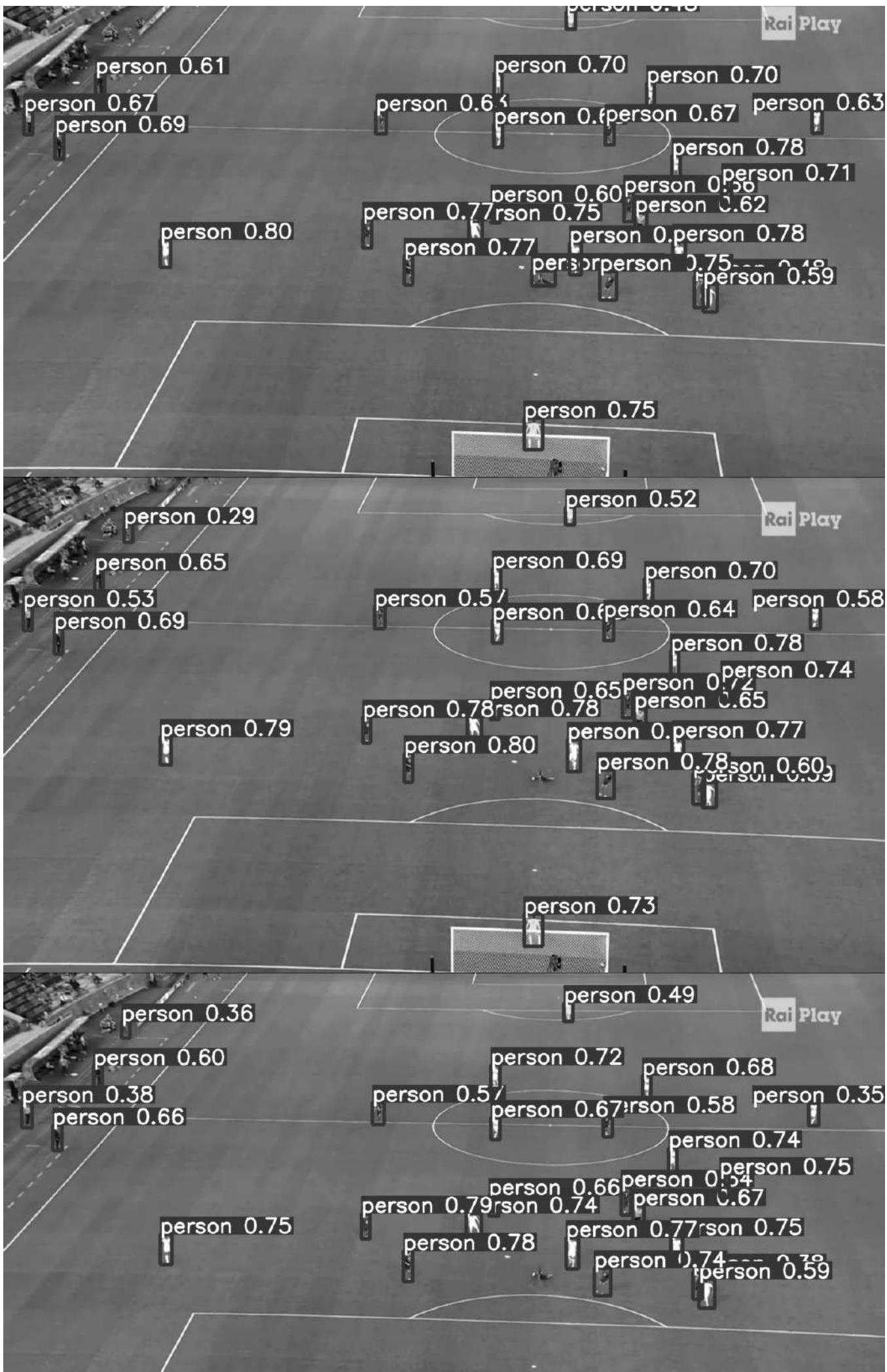


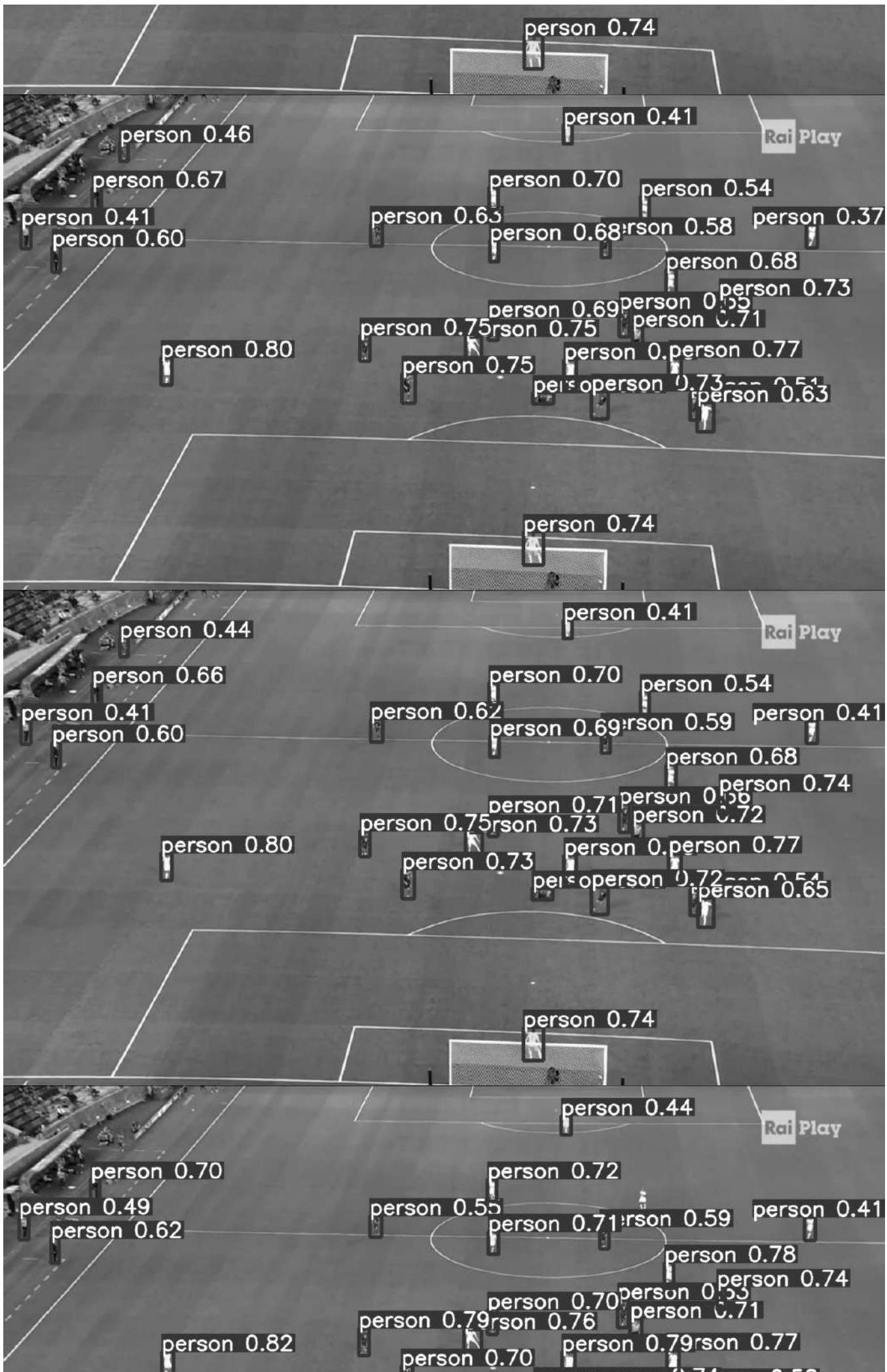


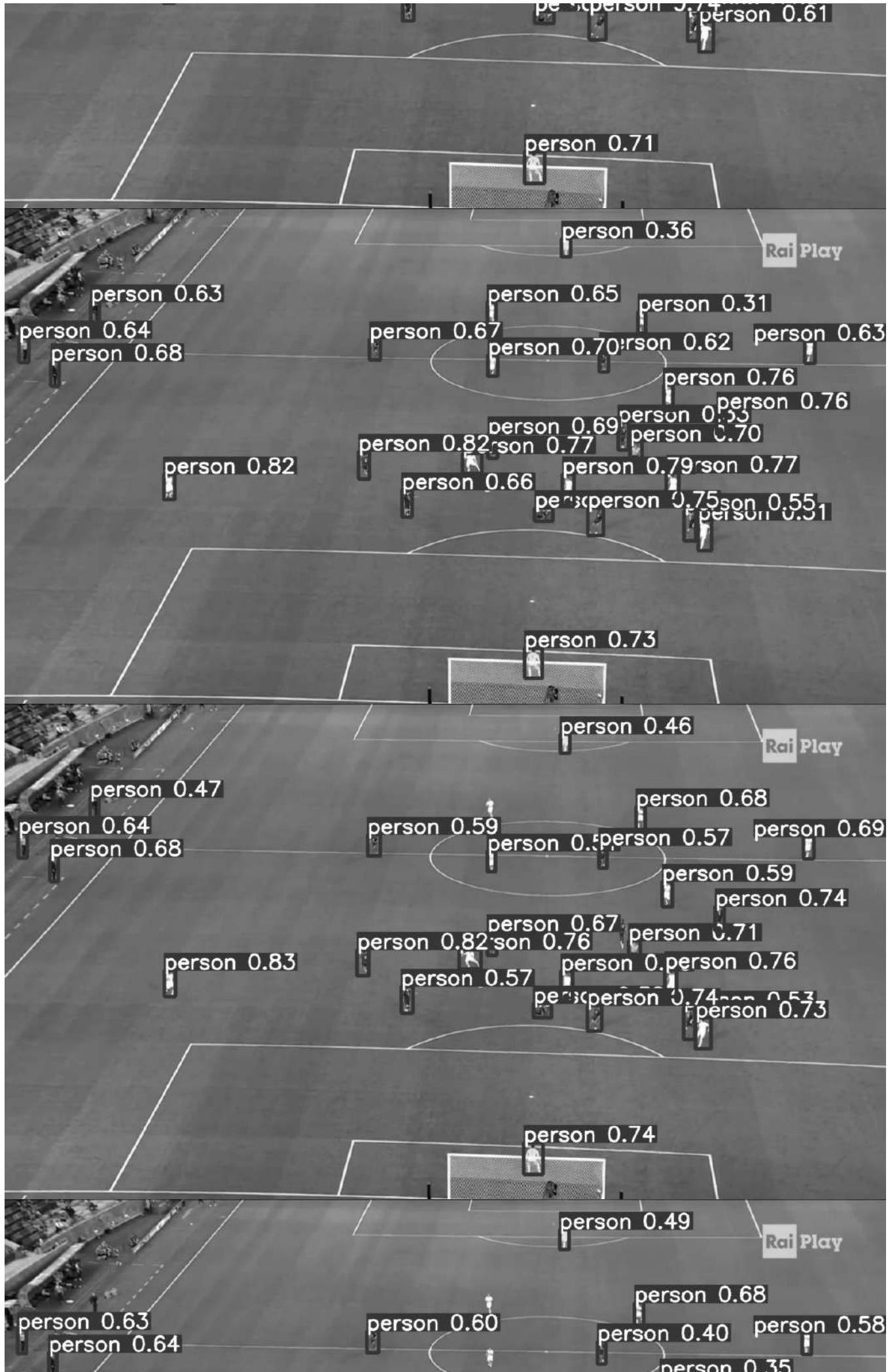






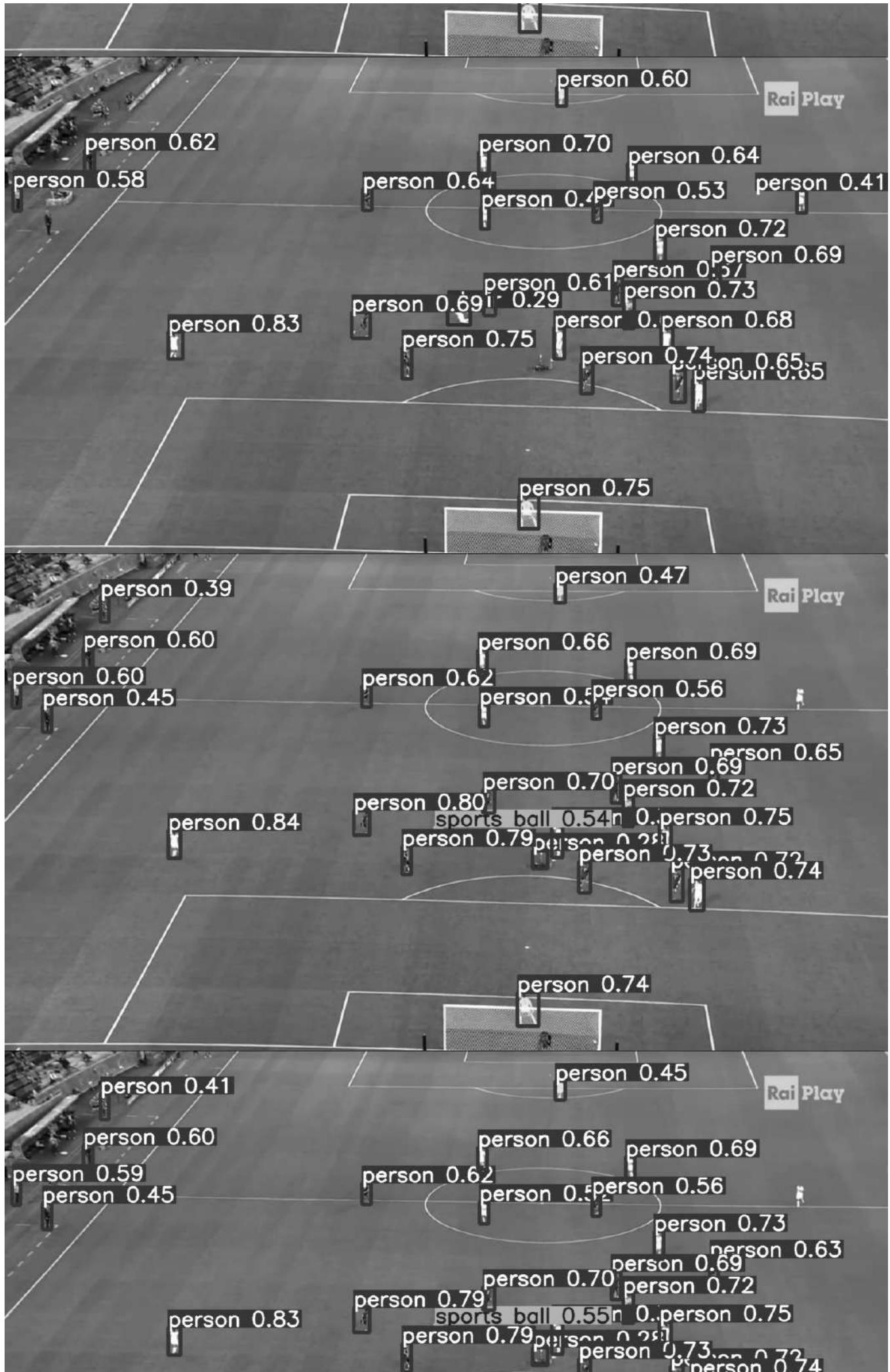


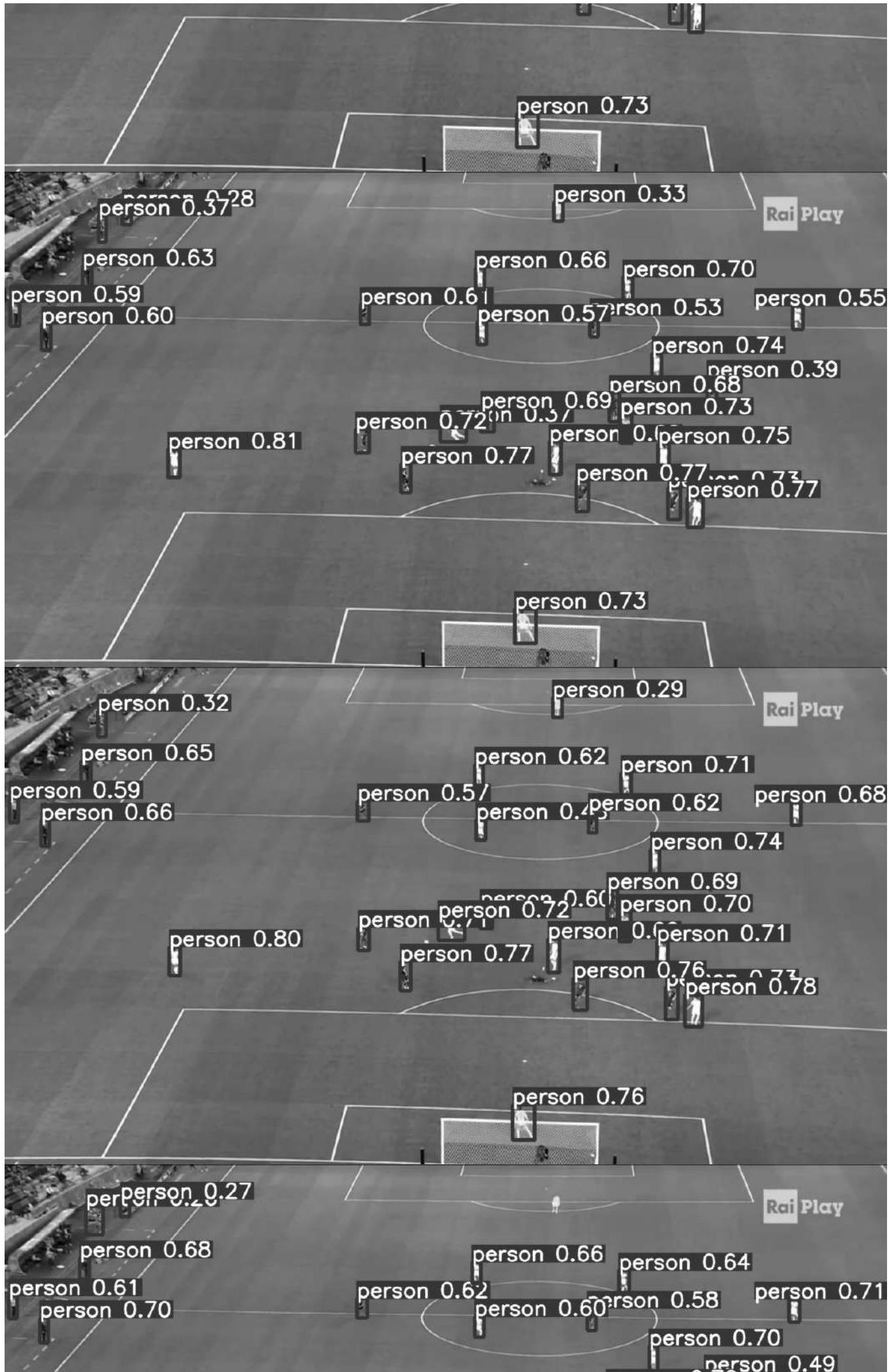




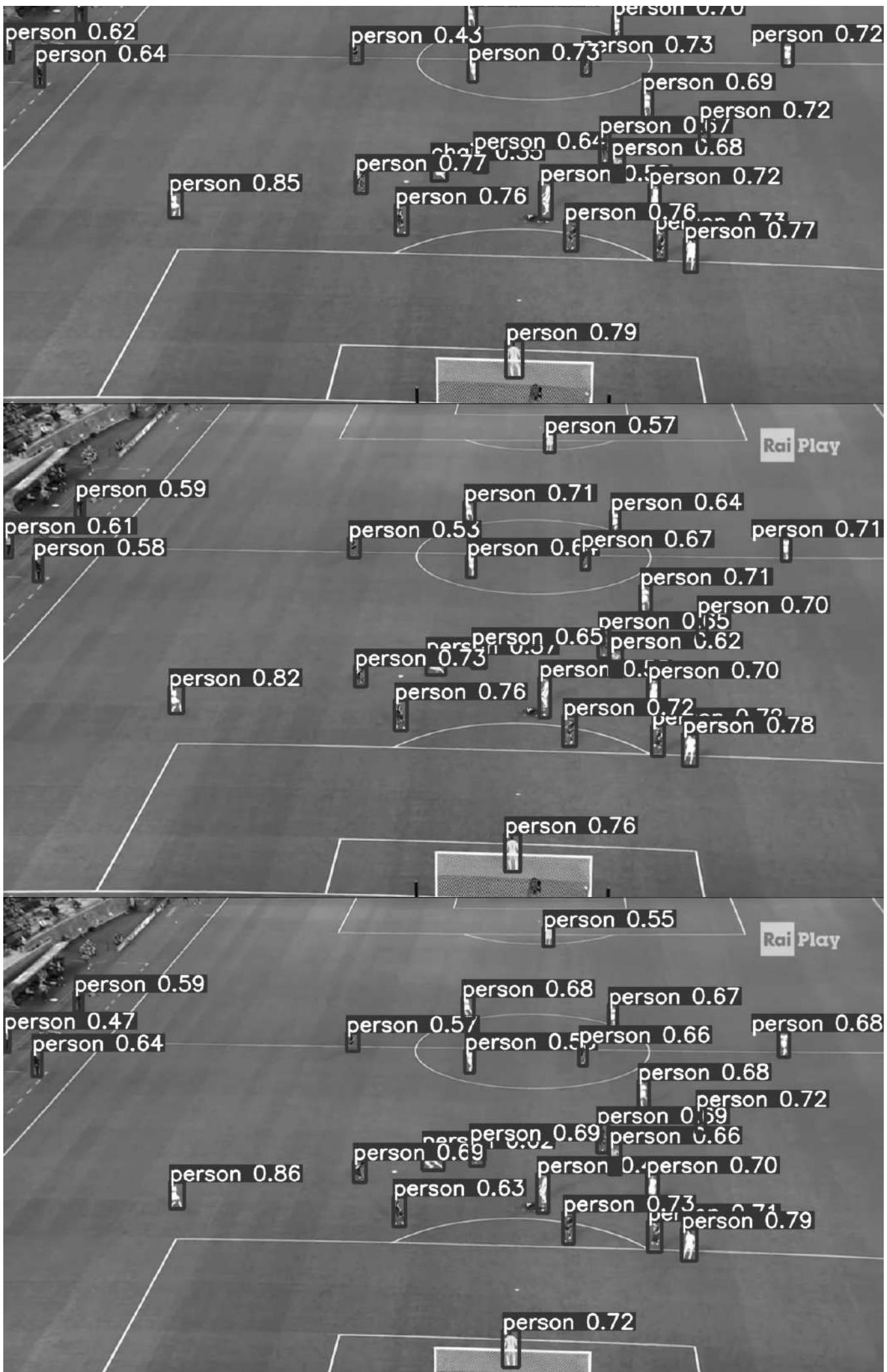




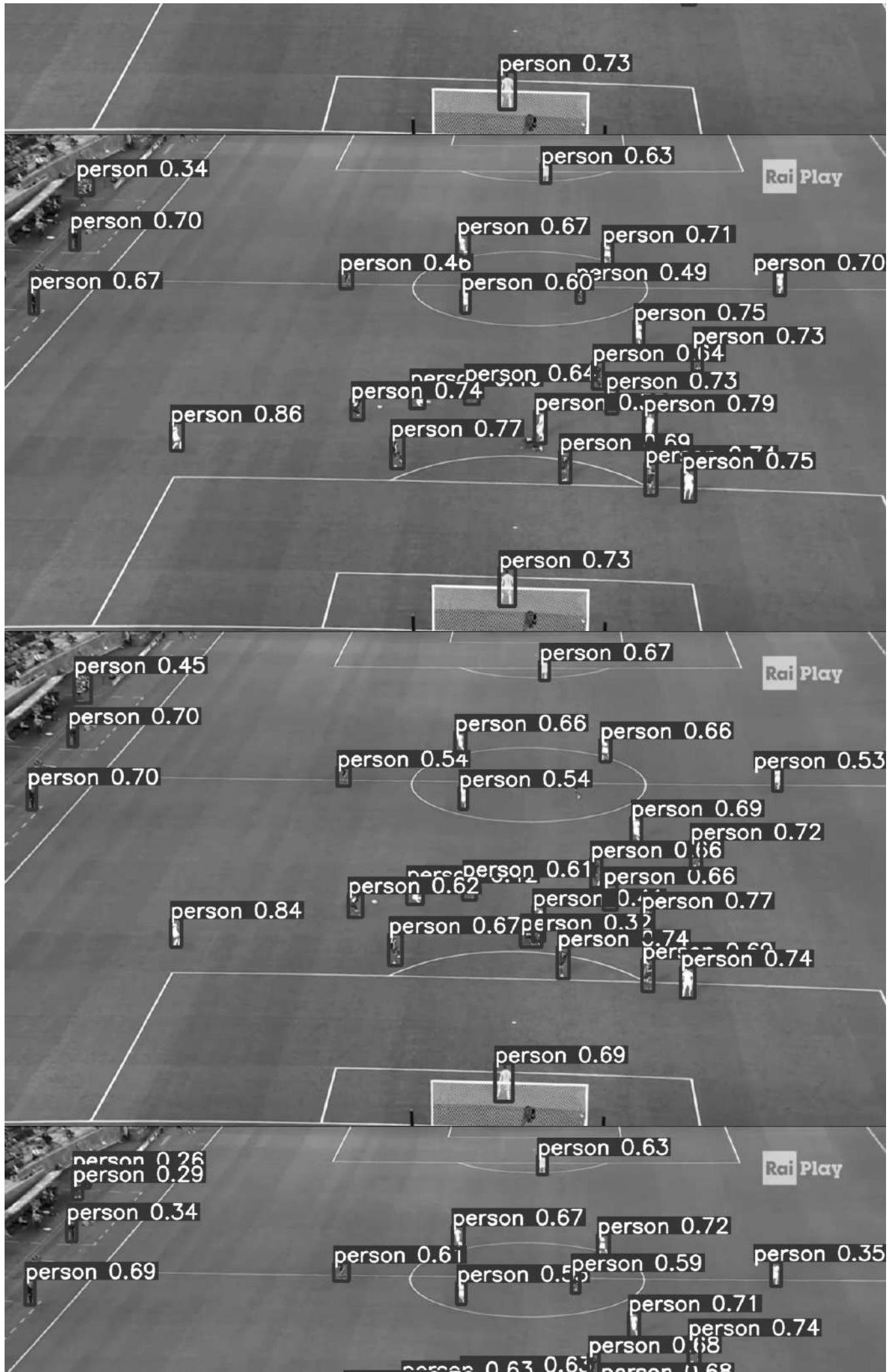


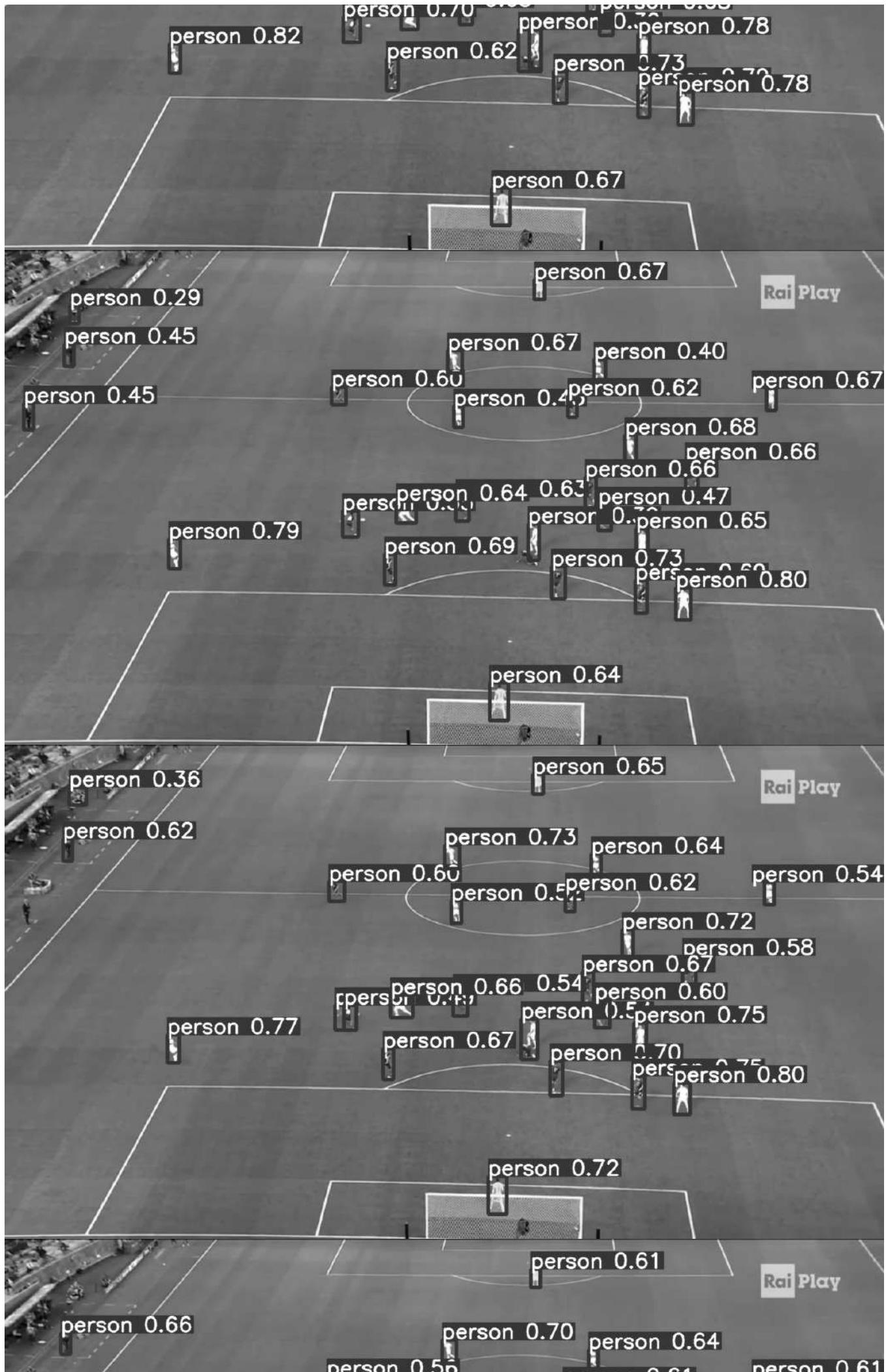


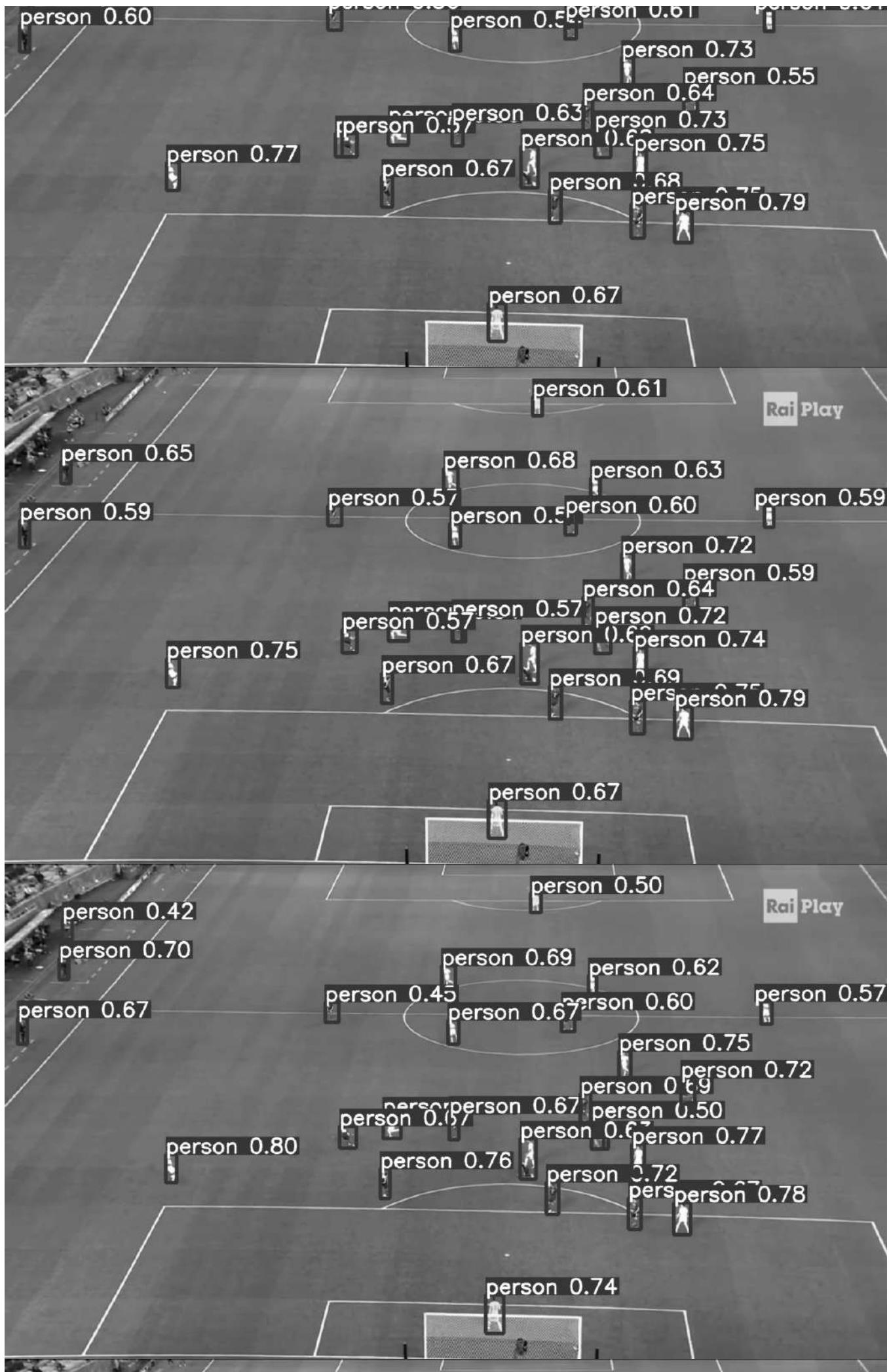


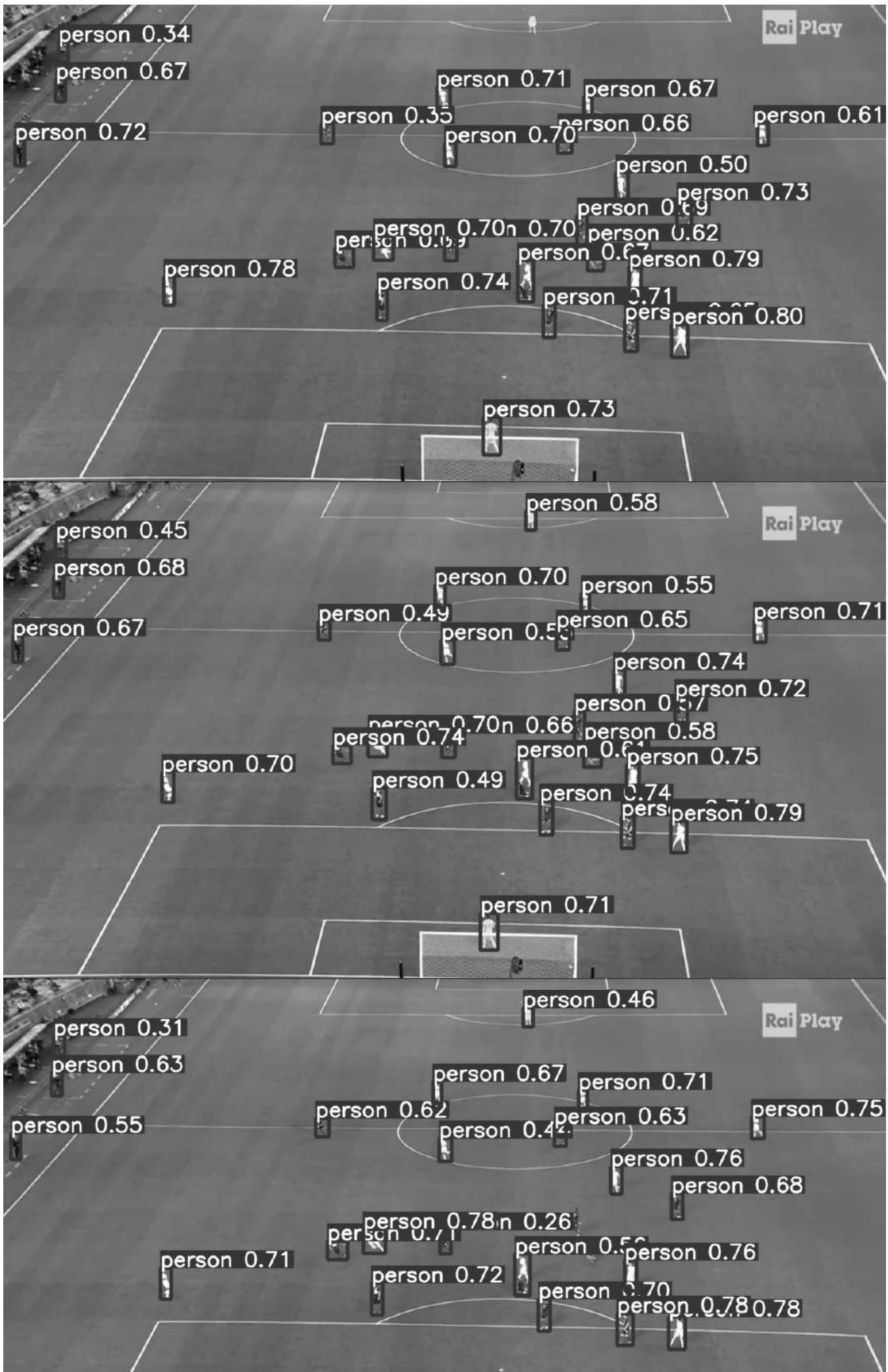




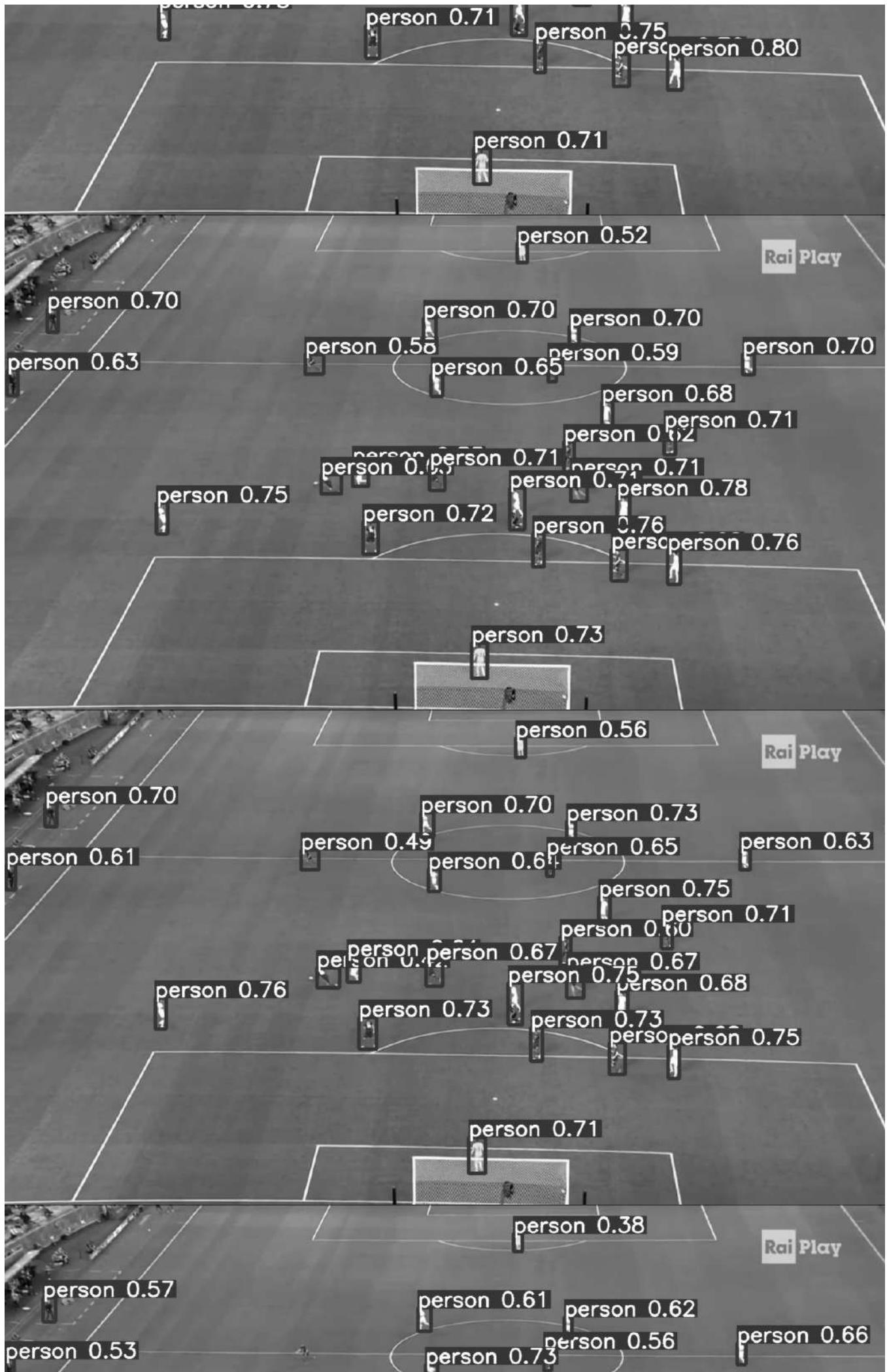


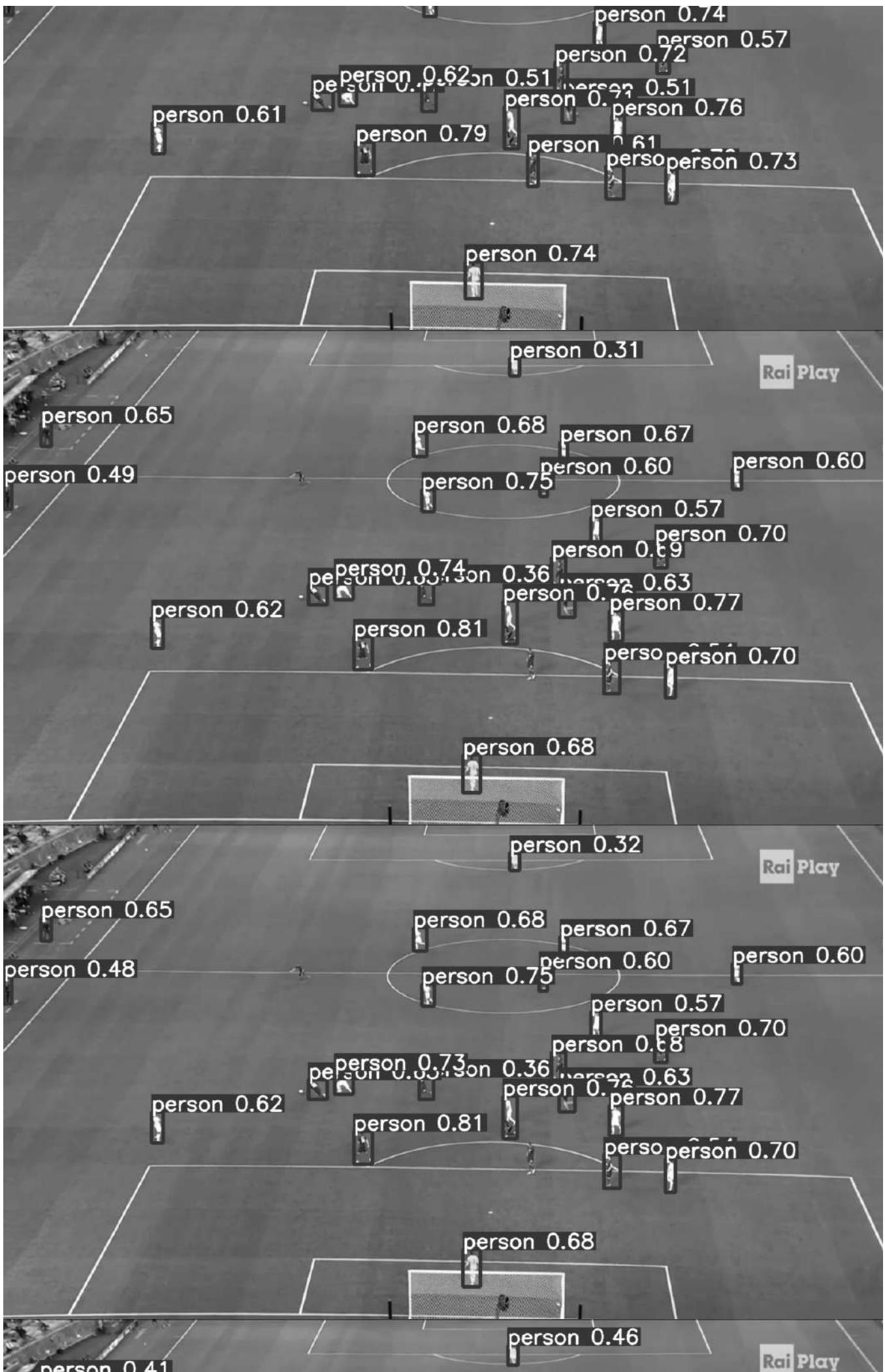


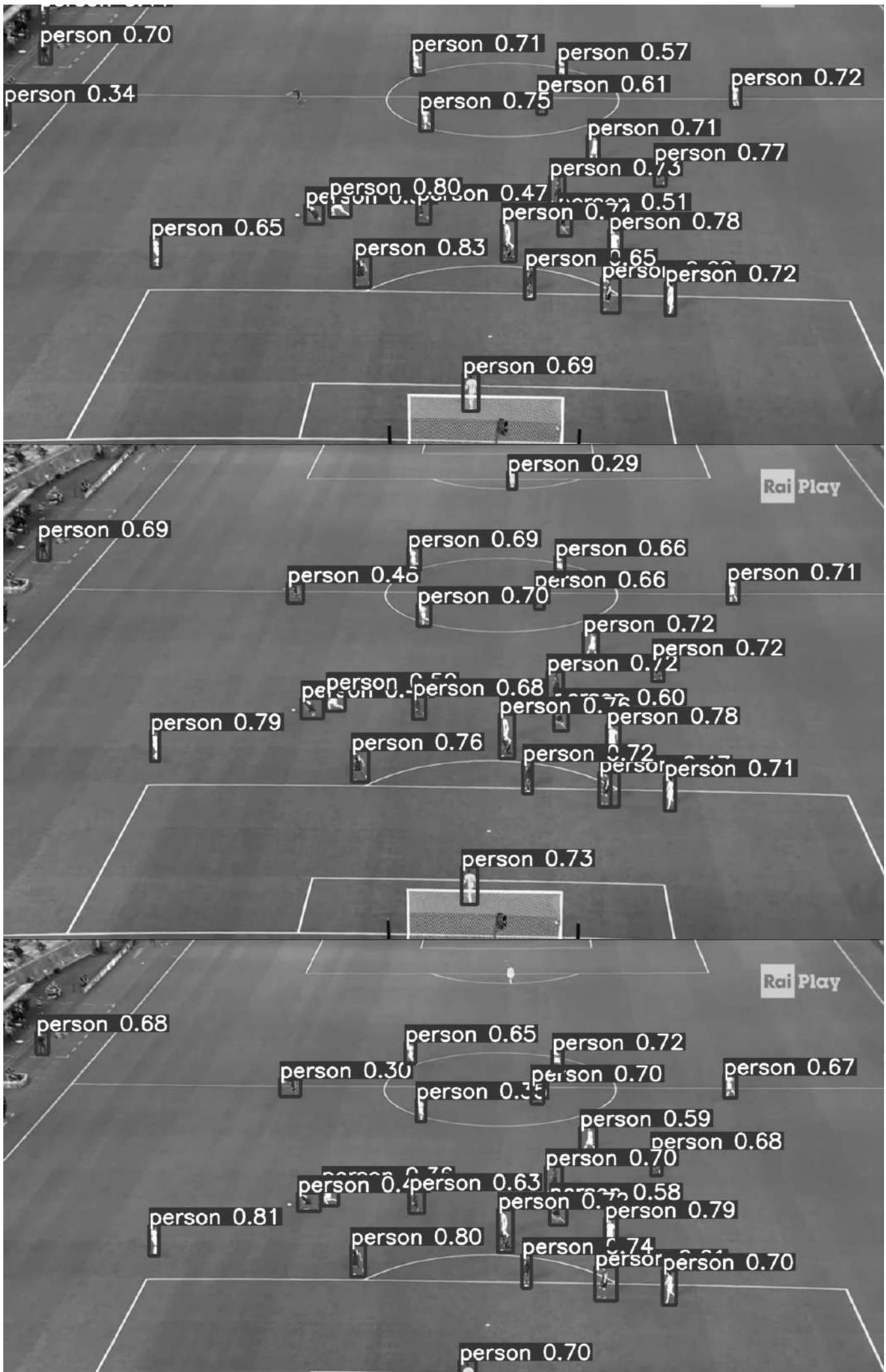


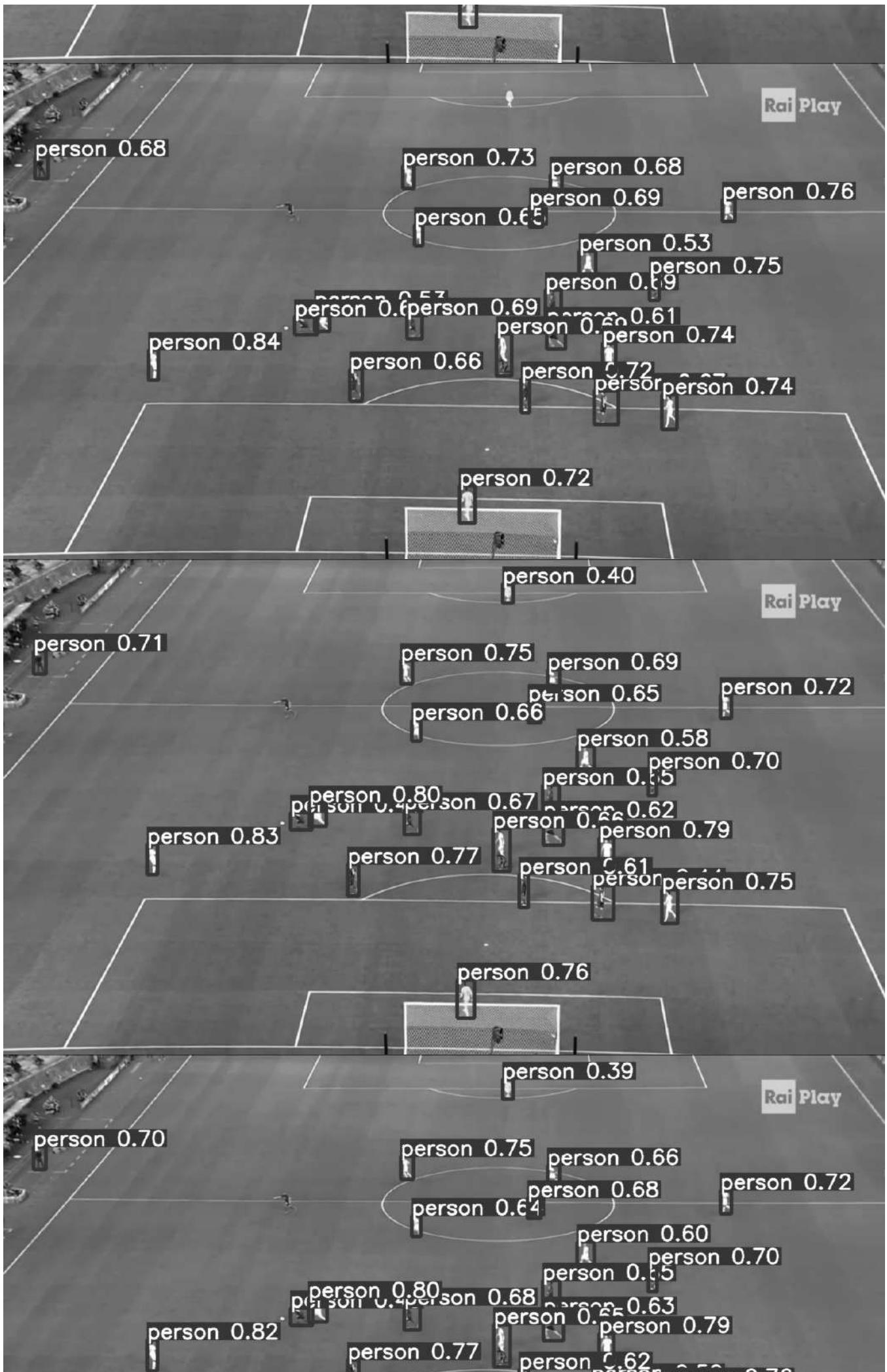


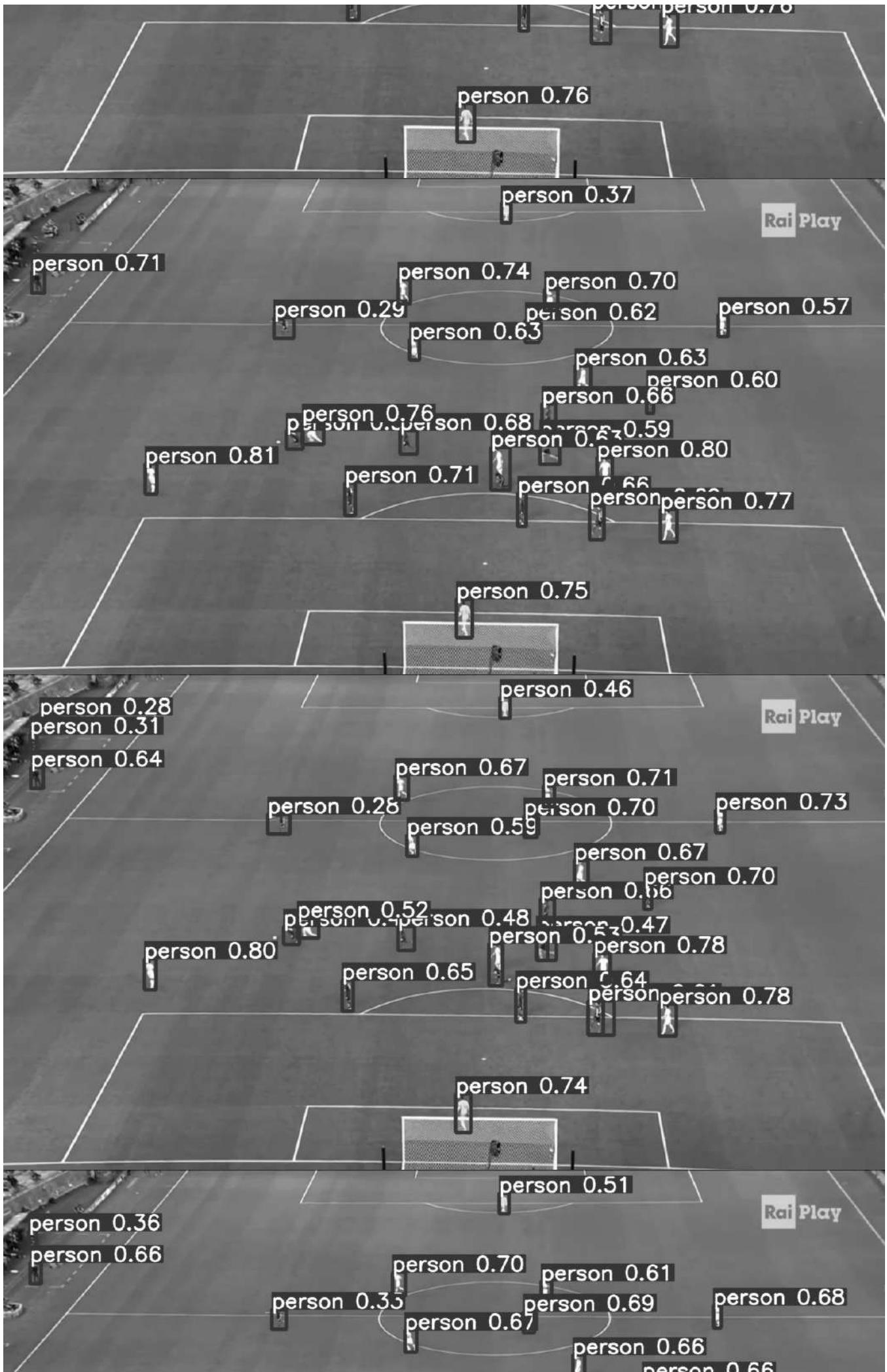


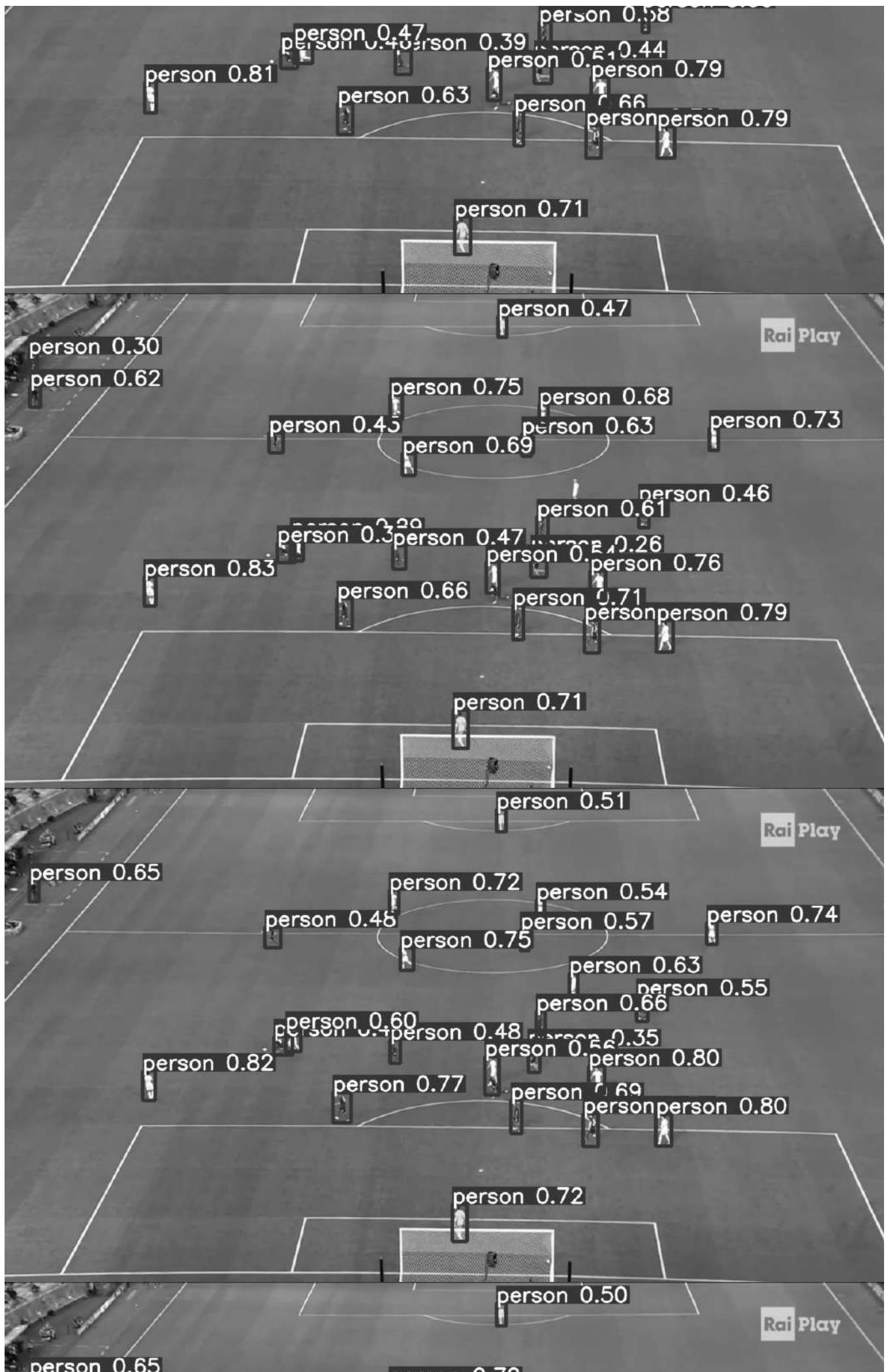


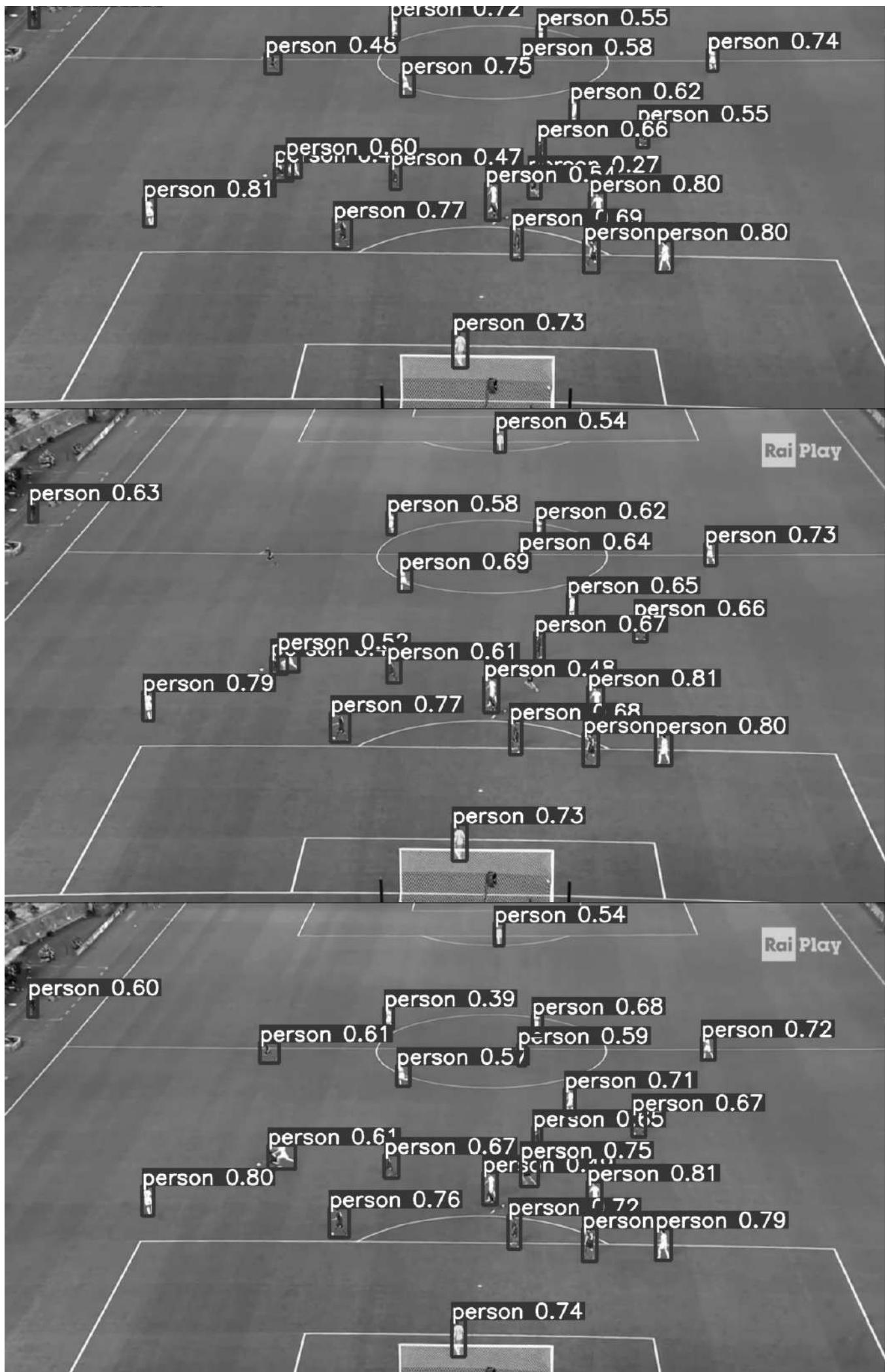


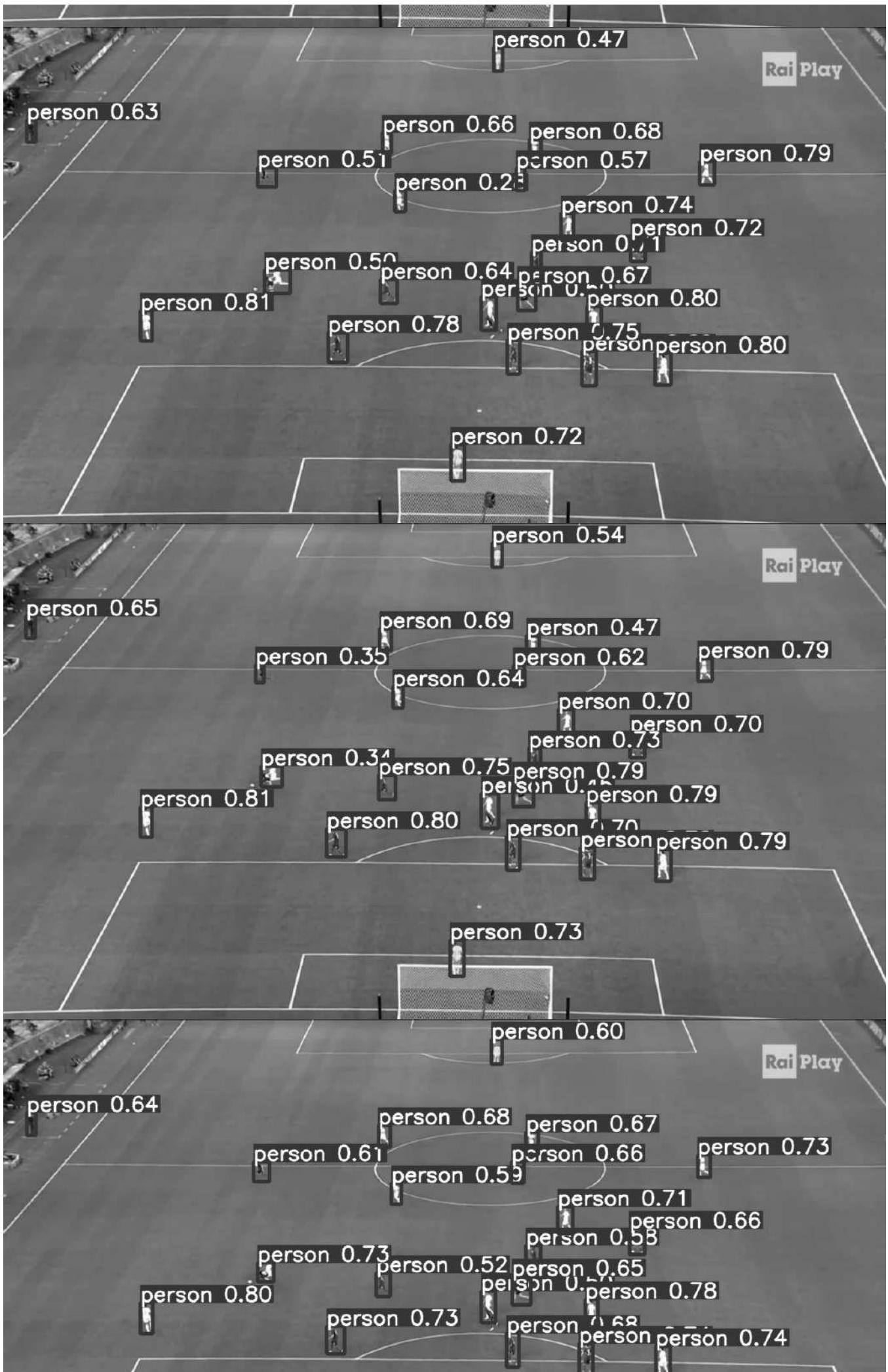


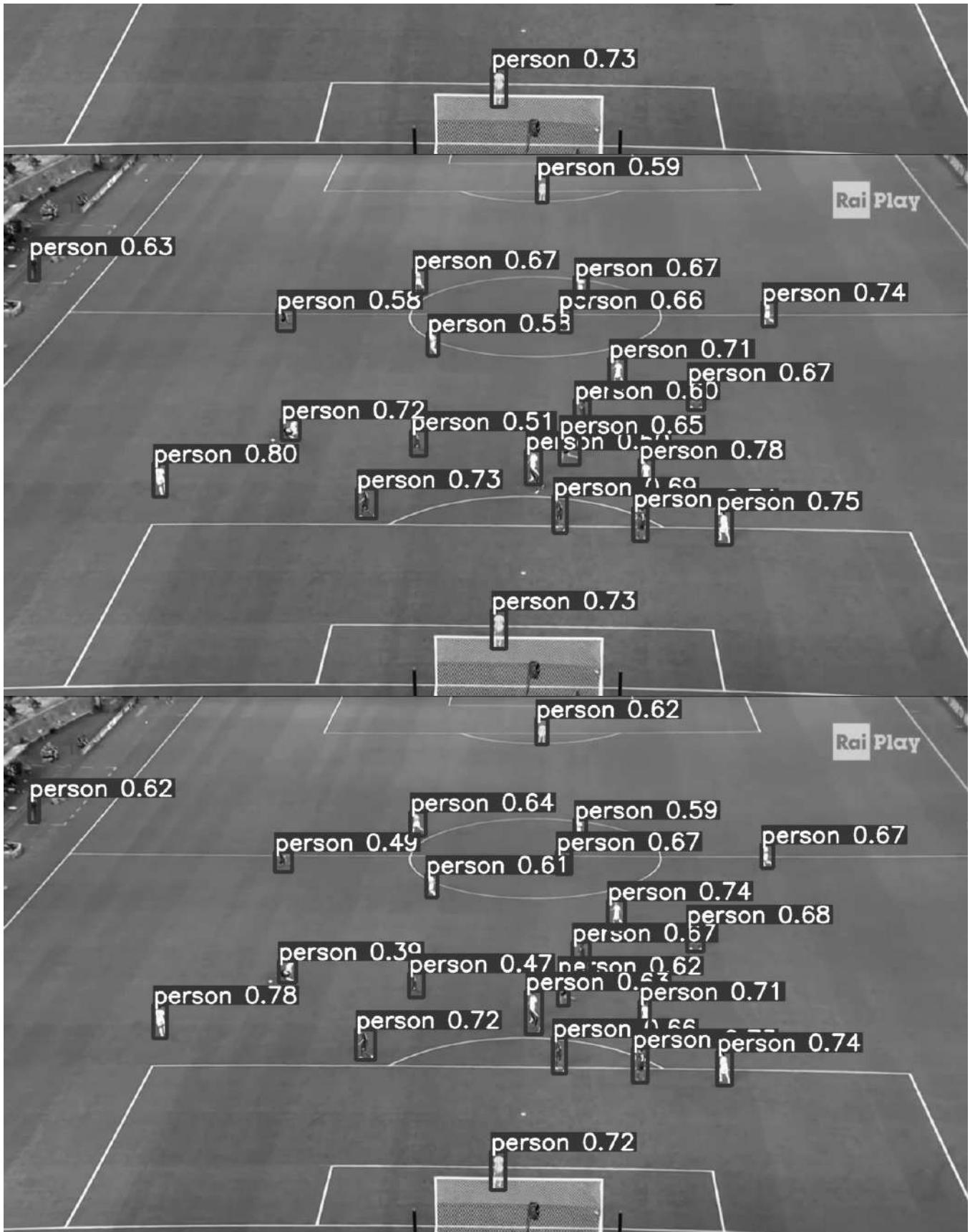












Start coding or [generate](#) with AI.